

EUROPEAN PATENT OFFICE

Patent Abstracts of Japan

PUBLICATION NUMBER : 09160814
PUBLICATION DATE : 20-06-97

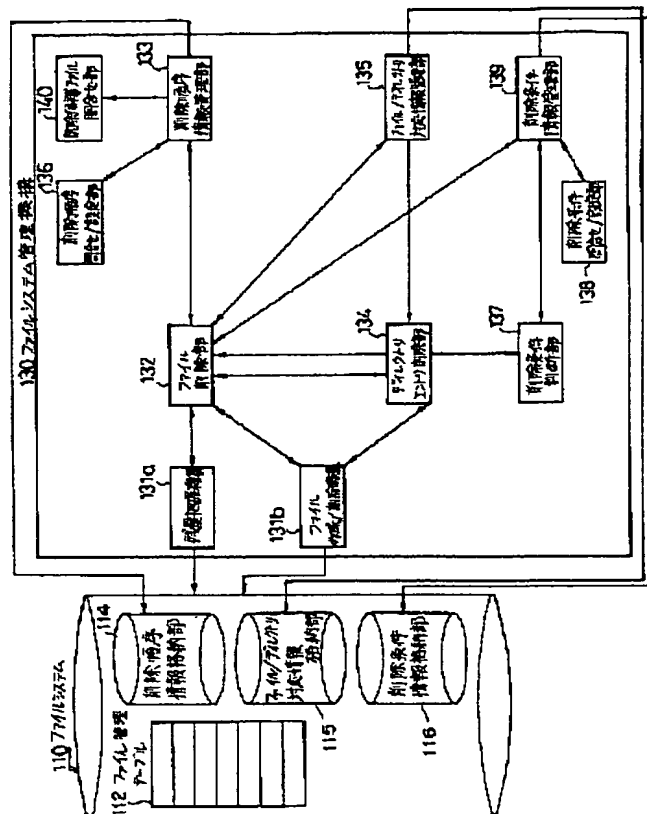
APPLICATION DATE : 04-12-95
APPLICATION NUMBER : 07315567

APPLICANT : FUJITSU LTD;

INVENTOR : ASANO KAZUO;

INT.CL. : G06F 12/00

TITLE : FILE MANAGEMENT SYSTEM



ABSTRACT : PROBLEM TO BE SOLVED: To provide a file management system whereby required free capacity can be secured on demand at a high-speed with a small processing quantity when the free capacity of a disk device becomes insufficient.

SOLUTION: A file system 110 consists of the disk device, etc., and is provided with a file control table 112 which stores information concerning respective files in every file as an entry. The residual amount grasping mechanism 131a of file system control mechanism 130 monitors the free area of a file storage area. When residual grasping mechanism 131a detects the shortage of free capacity, a file erasing part 132 inspects an erasure candidate file with a erasure order information managing part 133 and also refers to erasure condition information with an erasure condition information managing part 139 so as to erase the pre-existing file. At this time, file/directory correspondence information is obtained from a file/directory correspondence information managing part and the entry of the erased file is erased from the applying directory which is generated in a data block.

COPYRIGHT: (C)1997,JPO

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-160814

(43) 公開日 平成9年(1997)6月20日

(51) Int.Cl.⁶
G 0 6 F 12/00

識別記号 庁内整理番号
5 0 1

F I
G 0 6 F 12/00

技術表示箇所
5 0 1 H

審査請求 未請求 請求項の数23 O L (全 58 頁)

(21) 出願番号 特願平7-315567

(22) 出願日 平成7年(1995)12月4日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(72) 発明者 相川 秀幸

神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72) 発明者 浅野 一夫

神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(74) 代理人 弁理士 大曾 義之 (外1名)

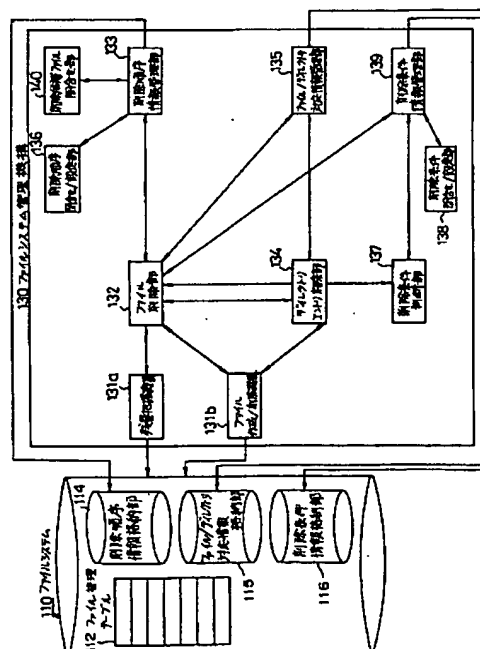
(54) 【発明の名称】 ファイルの管理システム

(57) 【要約】

【課題】 ディスク装置の空き容量が足りなくなった場合に、少ない処理量で、高速に、随時、必要なだけの空き容量を確保することができるファイル管理システムを提供する。

【解決手段】 ファイルシステム110は、ディスク装置等からなり、各ファイルに関する情報をファイル毎にエントリとして記憶するファイル管理テーブル112を有する。ファイルシステム管理機構130の残量把握機構131aはファイル記憶領域の空き領域を監視する。ファイル削除部132は、残量把握機構131aにより空き容量が足りなくなったことが検出されると、削除順序情報管理部133を介して削除候補ファイルを調べると共に、削除条件情報管理部139を介して削除条件情報を参照して、既成ファイルを削除する。この時、ファイル/ディレクトリ対応情報管理部からファイル/ディレクトリ対応情報を取得して、データブロック内に作成されている当該ディレクトリから上記削除されたファイルのエントリを削除する。

本発明の第一の基本構成を示すブロック図



【特許請求の範囲】

【請求項1】ファイルと該ファイルを管理するディレクトリが記憶媒体上に作成されるファイルシステムを管理するファイル管理システムにおいて、複数のエントリを有し、各エントリに前記記憶媒体上に作成されている各ファイルを管理するための情報を格納するファイル管理テーブルと、前記記憶媒体上に作成されているファイルとディレクトリの対応情報を管理するファイル／ディレクトリ対応情報管理手段と、前記記憶媒体のファイルの記憶容量の残量を把握する残量把握手段と、該記憶媒体に対してファイルの作成や削除を行うファイル作成／削除手段と、前記ファイル作成／削除手段が新規にファイルを作成する際に、前期残量把握手段によって把握されている残量情報を基に、前記記憶媒体にそのファイルを作成するための空き容量がないと判断した場合に、前記ファイル管理テーブルから前記記憶媒体上に既に作成されている既成ファイルのエントリを削除するファイル削除手段と、ファイル／ディレクトリ対応情報管理手段に管理されている情報を参照して、当該ディレクトリから、該ファイル削除手段によって削除されたファイルのエントリを削除するディレクトリエントリ削除手段と、を備えることを特徴とするファイル管理システム。

【請求項2】さらに、前記記憶媒体上に作成されている既成のファイルの削除順序に関する情報を管理する削除順序情報管理手段と、を有し、前記ファイル削除手段は、前記ファイル作成／削除手段が新規にファイルを作成する際に、前記記憶媒体上にそのファイルシステムを作成するための空き容量がなくなる毎に、前記削除順序情報管理手段から既成のファイルの削除順序に関する情報を取得し、そのファイル削除順序情報に従って前記ファイル管理テーブルのエントリに登録されている既成ファイル削除すると共に、該エントリを空にすること、を特徴とする請求項1記載のファイル管理システム。

【請求項3】さらに、前記削除順序情報管理手段に、前記ファイル管理テーブルのエントリに登録されているファイルの削除順序に関する情報を問い合わせ、該ファイルの削除順序に関する情報を取得する削除順序問い合わせ手段を、備えることを特徴とする請求項2記載のファイル管理システム。

【請求項4】さらに、前記削除順序情報管理手段に依頼して、前記ファイル管理テーブルのエントリに登録されているファイルの削除順序に関する情報を設定する削除候補設定手段手段を、備えることを特徴とする請求項2または3記載のファイル

管理システム。

【請求項5】さらに、前記削除順序情報管理手段に、削除候補となるファイルを問い合わせる削除候補ファイル問い合わせ手段を、備えることを特徴とする請求項2、3または4記載のファイル管理システム。

【請求項6】さらに、前記記憶媒体上に作成されているファイルの削除条件に関する情報を管理する削除条件情報管理手段と、ファイルを削除する際に、前記削除条件情報管理手段から前記ファイル管理テーブルのエントリに登録されている該ファイルの削除条件を取得し、その削除条件を基に、そのファイルを削除しても良いか否かを判断する削除条件判断手段を備え、前記ファイル削除手段は、該削除条件判断手段により削除しても良いと判断されたファイルのみを削除すること、を特徴とする請求項1または2記載のファイル管理システム。

【請求項7】さらに、前記削除条件情報管理手段に、前記ファイル管理テーブルのエントリに登録されているファイルの削除条件に関する情報を問い合わせ、そのファイルの削除条件情報を取得する削除条件問い合わせ手段を、備えることを特徴とする請求項6記載のファイル管理システム。

【請求項8】さらに、前記削除条件情報管理手段に依頼して、前記ファイル管理テーブルのエントリに登録されているファイルの削除条件を設定する削除条件設定手段を、備えることを特徴とする請求項6または7記載のファイル管理システム。

【請求項9】さらに、前記ファイル管理テーブルのエントリに登録されているN個の既成のファイルの削除順序に関する情報を、個々に管理するN個の削除順序情報管理手段と、前記N個の削除順序情報管理手段部を一元的に管理する削除順序情報総合管理手段とを備え、前記ファイル削除手段は、前記ファイル作成／削除手段が新規にファイルを作成する際に、記憶媒体にそのファイルシステムを作成するための空き容量がなくなる毎に、前記前記削除順序情報総合管理手段を介して、前記N個の削除順序情報管理手段によって管理されているN個の既成のファイルの削除順序情報を、上記優先順位順に取得し、その取得するファイル削除順序情報を基に、前記管理テーブルのエントリに登録されている既成ファイルを削除すると共に、該エントリを空にすること、を特徴とする請求項1記載のファイル管理システム。

【請求項10】さらに、前記N個の削除順序情報管理手段によって管理されてい

る前記N個の既成のファイルの削除順序情報の優先順位情報を管理する優先順位情報管理手段と、備え、前記削除順序情報総合管理手段は、前記優先順位情報管理手段を介して、前記優先順位情報管理手段によって管理されているN個の既成のファイルの削除順序の優先順位情報を取得して、その優先順位情報を基に、前記N個の削除順序情報管理手段によって管理されている前記N個の既成のファイル削除順序情報を上記優先順位順に取得すること、を特徴とする請求項9記載のファイル管理システム。

【請求項11】 さらに、前記優先順位情報管理手段に対して、前記N個の削除順序情報管理手段によって管理されている前記N個のファイル削除順序情報の優先順位を問い合わせる優先順位情報問い合わせ手段を、備えることを特徴とする請求項9または10記載のファイル管理システム。

【請求項12】 さらに、前記優先順位情報管理手段を介して、前記N個の削除順序情報管理手段によって管理されている前記N個のファイル削除順序情報の優先順位情報を設定する優先順位情報設定手段を、備えることを特徴とする請求項9、10または11記載のファイル管理システム。

【請求項13】 さらに、前記削除順序情報総合管理手段を介して、削除候補のファイルに関する情報を取得する削除候補問い合わせ手段を、備えることを特徴とする請求項9、10、11または12記載のファイル管理システム。

【請求項14】 前記ファイル管理テーブルは、各ファイル毎にそれぞれのファイルに関する情報を各エントリとして記憶しており、該エントリには、前記記憶媒体上に既に作成されているファイルあるいはディレクトリに関する情報が登録されている使用済のエントリ同士、及びまだいかなるファイルあるいはディレクトリに関する情報も登録されていない未使用のエントリ同士を、互いに論理的に接続する次エントリポインタを有することを特徴とする請求項1～13のいずれか1つに記載のファイル管理システム。

【請求項15】 前記ファイル管理テーブルの各エントリには、該エントリに登録されているファイルまたはディレクトリが属する上位のディレクトリを指すディレクトリポインタが設けられていることを特徴とする請求項14記載のファイル管理システム。

【請求項16】 個々のファイルのサイズ情報を有し、ファイルデータ記憶領域にファイルの実体を記憶して各ファイルの管理を行うファイル管理システムにおいて、ファイルデータ記憶領域中のデータの先頭位置を示す先頭位置保持手段と、

ファイルデータ記憶領域中のデータの最終位置を示す最終位置保持手段と、重ね書きの指示を行う重ね書き指示手段と、ファイルデータ記憶領域にデータを追記する際に、前記重ね書き指示手段からデータの重ね書きの指示がなされた場合、ファイルデータ記憶領域の先頭から、該ファイルのデータを上書きするデータ書き込み手段と、を備えたことを特徴とするファイル管理システム。

【請求項17】 個々のファイルのサイズ情報を有し、ファイルデータ記憶領域にファイルの実体を記憶して各ファイルの管理を行うファイル管理システムにおいて、ファイルデータ記憶領域中のデータの先頭位置を示す先頭位置保持手段と、

ファイルデータ記憶領域中のデータの最終位置を示す最終位置保持手段と、

重ね書きの指示を行う重ね書き指示手段と、ファイルサイズの上限值を設定するサイズ上限値保持手段と、

ファイルの現状サイズとファイルサイズの上限值とを比較する比較手段と、

ファイルデータ記憶領域へのデータの追加書き込み中に、前記比較手段によって、ファイルサイズがファイルサイズの上限值に等しくなったことが検出されたとき、前記重ね書き指示手段からの指示を受けて、自動的にファイルの先頭からデータを上書きする動作を開始するデータ書き込み手段を、備えたことを特徴とするファイル管理システム。

【請求項18】 複数種類の単位を用いてファイルサイズの上限值が指定されたとき、該単位を解析する単位解析手段と、

該単位解析手段の解析結果を基に、ファイルサイズの上限值を、予め定められた所定の単位で前記サイズ上限値保持手段に設定する設定手段と、

をさらに備えたことを特徴とする請求項17記載のファイル管理システム。

【請求項19】 個々のファイルのサイズ情報を有し、ファイルデータ記憶領域にファイルの実体を記憶して各ファイルの管理を行うファイル管理システムにおいて、ファイルシステム全体の空き容量の下限值が設定される空き容量下限保持手段と、

ファイルシステムの空き容量が該空き容量下限保持手段に保持されている値より少なくなったことを判断する空き容量判定手段と、

該空き容量判定手段によって、ファイルシステムの空き容量が空き容量下限保持手段の値より少なくなったと判断されたとき、ファイルに対してデータを重ね書きするように指示する重ね書き指示手段を、

備えることを特徴とするファイル管理システム。

【請求項20】 前記空き容量判定手段によってファイルシステムの空き容量が空き容量下限保持手段に保持され

ている値より少なくなったと判断されたとき、自動的に重ね書きを行って良いかどうかを示す自動重ね書きフラグを、個々のファイル毎に設け、前記重ね書き指示手段は、前記空き容量判定手段によって、あるファイルの空き容量が空き容量下限保持部に保持されている値より少なくなったと判断されたとき、そのファイルの自動重ね書きフラグを参照して、そのフラグに重ね書きが可能と設定されていた場合にのみ、そのファイルにデータを重ね書きするように指示することを、特徴とする請求項19記載のファイル管理システム。

【請求項21】ファイルに格納されている各データのデータ構造を記憶するデータ構造情報記憶手段と、前記データ構造情報記憶手段に格納されている情報を参照して、ファイルに書き込まれている各データのデータ構造を検出し、各データの区切り位置を検出するデータ境界検出手段を更に備え、前記データ境界検出手段は、ファイルにデータが上書きされる際に、上書きされているデータのデータ構造が損なわれないような先頭位置を、前記先頭位置保持手段に設定することを、特徴とする請求項17、18、19、または20に記載のファイル管理システム。

【請求項22】前記データ境界検出手段は、あるデータが上書きされた際、そのデータの次に書き込まれているデータの先頭位置を、前記先頭位置保持手段に設定することを特徴とする請求項21に記載のファイル管理システム。

【請求項23】ファイルに書き込まれる各データの先頭には、データの区切り位置を示すある特定のレコードが書かれており、前記データ境界検出手段は、この特定のレコードを検出することにより、各データの先頭位置を検出することを中心とする請求項21または22に記載のファイル管理システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ファイルシステムの管理方式に関し、特に、ファイルの空き容量が不足した場合におけるファイルの管理システムに関する。

【0002】

【従来の技術】コンピュータシステムでは、データの記憶装置として磁気ディスク装置や光ディスク装置が、主に、用いられている。このような記憶装置においては、データはファイルによって管理される。

【0003】ところで、最近、インターネットが急速に普及している。このインターネット上のサービスの一つに、一般に、ニュースと呼ばれている電子掲示板というものがある。このニュースには、現在、約数千のニュースグループがあり、多種多様な記事が書き込まれ、多数のユーザに閲覧されている。

【0004】このインターネット上で流通する電子掲示

板の記事の量は、日に日に増加しており、最近ではそのデータ量は150～200MB（メガバイト）にも達している。したがって、これらの記事を受信・管理しているサイトでは、記事を保存するためのスプール容量の確保が重要な課題となっている。

【0005】この電子掲示板の仕組みは、一般に、以下のようになっている。受信される記事は、いったん、ディスク装置上に設けられたスプール領域に保管される。このスプール領域の容量には制限があるため、一定期間、経過した記事は、順次、削除されていく。この場合、古い記事から順次、削除されていく。この場合、記事が削除されるまでの期間（保存期間）は、用意されたスプール領域の容量と記事の流通量、及び、記事を受信・提供しているサイトにおける記事の重要性などの要因の兼ね合いによって決定される。このとき、記事の保存期間は、常に、一定であり、かつ、スプール領域が過多になったときの記事の削除の操作は定期的／自動的に行われるのが一般的である。

【0006】ところで、記事の流通量は日によってばらつきがあるのが普通である。このとき、そのばらつきが平均値よりも大幅に多い日が連続すると、確保していたスプール領域に記事が入りきらなくなる。このような場合、電子掲示板のシステムやホスト自体が障害を起こすこともある。

【0007】このため、従来は、

① ディスク装置の空き容量を監視し、空き容量が少なくなってきたら、記事の保存期間のパラメータを短く変更し、古い記事を削除する。

【0008】② 上記削除の結果、必要と考えられるだけの空き容量が確保できれば、削除処理を停止し、確保できなければ、確保できるまで、同様の操作を繰り返す。という方法を人手により行っていた。

【0009】また、計算機システムにおいては、何らかのサービスを提供している場合、その利用状況などのログをある程度、ファイルに記録・保存しておき、障害が発生した場合、そのログを解析して障害原因を究明する処理が、一般に、行われる。

【0010】ところが、ログの量は時間とともに増加する一方で、不注意に運用していると、ログファイル自体がファイルシステムの空き容量の不足を引き起こし、本来のサービスの運用に支障を来す場合がある。

【0011】このため、従来は、ログファイルの名前を、例えば、定期的に（普通は一日～一週間）、FILE.0～FILE.7など一つづつ変えていき、ある程度古いファイルを自動的に削除していくという方法を取っていた。

【0012】

【発明が解決しようとする課題】しかしながら、上記従来の電子掲示板の方法では、以下のような問題があった。

【0013】① ディスク装置の空き容量の監視を、頻繁に行うのは手間が大変である。

② 例え、ディスク装置の空き容量の監視を、ある程度、自動化して頻繁に行ったとしても、記事の保存期間のパラメータの短縮の度合いの判断は自動化しにくい。このため、記事の削除は人手で行うのが一般的とならざるをえない。しかし、土曜日、日曜日などは、必ず、管理者がいるとは限らないので、結局、タイムリーな削除はできない。

【0014】③ また、例え、うまくタイムリーに記事を削除できたとしても、記事の削除処理には時間がかかる。このため、長時間、記事の受信処理が止まったり、記事の削除のために、無駄にCPU資源を浪費する。

【0015】また、上記従来のログファイルの方法では、以下のような問題があった。

① ログファイルに記録されるログの量が、毎日、一定していないため、突出してログの多い日が続くことがあると、ファイルシステムの空き容量が不足してしまう。

【0016】② 上記①のような事態を防止するために、ログファイルに最低限のログ量だけしか保存しないようにすると、実際に障害が発生した場合、その障害の原因の究明に必要なログを記録しそこねる場合がある。本発明は、ディスク装置の空き容量が足りなくなった場合に、少ない処理量で、高速に、随時、必要なだけの空き容量を確保することができるようにすることを目的とする。また、常に、一定量のログを保存することができ、ファイルシステムの空き容量を圧迫することなしに、ログを保存できるようにすることを目的とする。

【0017】

【課題を解決するための手段】本発明のファイル管理システムは、ファイルと該ファイルを管理するディレクトリが記憶媒体上に作成されるファイルシステムを管理するファイル管理システムを前提とする。

【0018】そして、複数のエントリを有し、各エントリに前記記憶媒体上に作成されている各ファイルを管理するための情報を格納するファイル管理テーブルと、前記記憶媒体上に作成されているファイルとディレクトリの対応情報を管理するファイル/ディレクトリ対応情報管理手段と、前記記憶媒体のファイルの記憶容量の残量を把握する残量把握手段と、該記憶媒体に対してファイルの作成や削除を行うファイル作成/削除手段と、前記ファイル作成/削除手段が新規にファイルを作成する際に、前期残量把握手段によって把握されている残量情報を基に、前記記憶媒体にそのファイルを作成するための空き容量がないと判断した場合に、前記ファイル管理テーブルから前記記憶媒体上に既に作成されている既成ファイルのエントリを削除するファイル削除手段と、ファイル/ディレクトリ対応情報管理手段に管理されている情報を参照して、当該ディレクトリから、該ファイル削除手段によって削除されたファイルのエントリを削除す

るディレクトリエントリ削除手段とを備える。

【0019】記憶媒体に対しては、アプリケーション等からの指示に従いファイル作成/削除手段が、ファイルの作成/削除を行う。ファイルが作成される度に、この作成されたファイルに関する情報がファイル管理テーブルのエントリに、各ファイル毎に個別に登録される。残量把握手段は、常に記憶媒体の空の記憶領域の量を監視する。残量把握手段が空の記憶領域が無くなったことを検出した場合には、ファイルの削除をファイル削除手段に依頼する。ファイル削除手段は、その削除を依頼されたファイルを記憶媒体上から削除すると共に、ファイル管理テーブルのその削除されたファイルのエントリを空にする。そして、ファイルを削除することによって空いた記憶領域に新たにファイルを作成し、このファイルに関する情報をファイル管理テーブルの空きとなっているエントリに登録する。

【0020】また、ファイルの記憶形態がディレクトリ構造を有している場合には、記憶媒体上にディレクトリ（ディレクトリファイル）が作成されている。このディレクトリの各エントリには、例えば、そのディレクトリに属するファイルまたはディレクトリの名称とデータブロックへのポインタが格納されている。従って、あるディレクトリの下にあるファイルを削除する場合に、ファイルを削除するだけでは、このファイルが属するディレクトリにそのファイルのエントリが残されてしまう。そこで、ファイル/ディレクトリ対応情報管理部がファイル/ディレクトリ対応情報を参照して、ディレクトリエントリ削除部に、当該ディレクトリのエントリからそのファイルのエントリを削除させる。

【0021】本発明のファイル管理システムの他の構成においては、個々のファイルのサイズ情報を有し、ファイルデータ記憶領域にファイルの実体を記憶して各ファイルの管理を行うファイル管理システムを前提とする。そして、ファイルデータ記憶領域中のデータの先頭位置を示す先頭位置保持手段と、ファイルデータ記憶領域中のデータの最終位置を示す最終位置保持手段と、重ね書きの指示を行う重ね書き指示手段と、ファイルデータ記憶領域にデータを追記する際に、前記重ね書き指示手段からデータの重ね書きの指示がなされた場合、ファイルデータ記憶領域の先頭から、該ファイルのデータを上書きするデータ書き込み手段と、を備える。

【0022】先頭位置保持手段、及び最終位置保持手段は、ファイルデータ記憶領域中のファイルが記憶されている領域の先頭位置と最終位置を示すポインタを保持する。ファイルデータ記憶領域にデータが書き込まれていき、所定の領域内に納まらなくなると、重ね書き指示手段はデータ書き込み手段にファイルデータ領域の先頭からファイルの重ね書きを指示する。この際、直接、重ね書きをすると重ね書きされたデータ（特に、データの先頭部分にはデータ構造に関する重要な情報が組み込まれ

ていることが多い)は失われてしまうために、ファイルシステムがデータを読み取ろうとするとエラーを起こしてしまう可能性がある。そこで、先頭位置保持手段が保持する最古のデータの先頭位置を、重ね書きされているデータの構造に応じて、変化させることが可能である。すなわち、最終位置保持手段に最新のデータ最終位置を保持させ、先頭位置保持部に最古のデータの先頭位置を保持させるようにする。このことにより、各ファイルについて、データの先頭位置と最終位置を正確に把握することが可能になる。また、ファイルシステムがデータの読み込み動作を行う場合にエラーを起こすことがなくなる。

【0023】また、本発明のファイル管理システムの更に他の構成においては、個々のファイルのサイズ情報を有し、ファイルデータ記憶領域にファイルの実体を記憶して各ファイルの管理を行うファイル管理システムを前提とし、ファイルデータ記憶領域中のデータの先頭位置を示す先頭位置保持手段と、ファイルデータ記憶領域中のデータの最終位置を示す最終位置保持手段と、重ね書きの指示を行う重ね書き指示手段と、ファイルサイズの上限值を設定するサイズ上限値保持手段と、ファイルの現状サイズとファイルサイズの上限值とを比較する比較手段と、ファイルデータ記憶領域へのデータの追加書き込み中に、前記比較手段によって、ファイルサイズがファイルサイズの上限值に等しくなったことが検出されたとき、前記重ね書き指示手段からの指示を受けて、自動的にファイルの先頭からデータを上書きする動作を開始するデータ書き込み手段を備える。

【0024】このように、ファイルのサイズの上限值をサイズ上限値保持手段に設定しておき、比較手段によって現状のファイルサイズと予め設定されたファイルサイズの上限值とを比較することによって、データを追記しているファイルのサイズがそのファイルのサイズの上限值に等しくなった場合には、ファイルの先頭位置から追記データの重ね書きを行う。

【0025】したがって、個々のファイルのサイズを、一定容量に制限できるので、ファイルシステム全体の空き容量を圧迫しないような運用が可能となる。本発明のファイル管理システムの更に別の構成においては、個々のファイルのサイズ情報を有し、ファイルデータ記憶領域にファイルの実体を記憶して各ファイルの管理を行うファイル管理システムを前提とし、ファイルシステム全体の空き容量の下限値が設定される空き容量下限保持手段と、ファイルシステムの空き容量が該空き容量下限保持手段に保持されている値より少なくなったことを判断する空き容量判定手段と、該空き容量判定手段によって、ファイルシステムの空き容量が空き容量下限保持手段の値より少なくなったと判断されたとき、ファイルに対してデータを重ね書きするように指示する重ね書き指示手段を備える。

【0026】この構成においては、ファイルシステムの全体の空き容量の下限値を設定することにより、常に、ファイルシステムの全体の空き容量を一定量確保することができ、ファイルシステム全体の空き容量を圧迫しないような運用が可能となる。

【0027】

【発明の実施の形態】以下、図面を参照しながら、本発明の実施例を説明する。図1は、本発明の第一の基本構成を示すブロック図である。

【0028】ファイルシステム110は、ディスク装置上に作成され、データが格納されるファイル、ファイル名とそのファイルの実体に関する情報が格納されているiノードの番号が対になっているディレクトリファイル、ファイル情報を管理するファイル管理テーブル112、ファイルの削除順序に関する情報を格納する削除順序情報格納部114、ファイルシステム110内のファイルとディレクトリエントリとの対応情報を格納するファイル/ディレクトリエントリ対応情報格納部115、ファイルシステム110内のファイルの削除条件に関する情報を格納している削除条件情報格納部116等を備えている。尚、上記削除順序情報格納部114、上記ファイル/ディレクトリエントリ対応情報格納部115及び上記削除条件情報格納部116は、ファイルシステム11とは別の記憶装置に設けるようにしてもよい。

【0029】ファイルシステム管理機構130は、ファイルシステム110の空き容量を監視する残量把握機構131aとファイルシステム110にファイルを作成したり、削除したりするファイル作成/削除機構131bを備えている。また、ファイルシステム管理機構130は、ファイル作成/削除機構131bに依頼して上記ファイルシステム110内のファイルを削除するファイル削除部132、該ファイル削除部132が削除するファイルの順序に関する情報を管理する削除順序情報管理部133、上記ファイルシステム110内のディレクトリエントリを削除するディレクトリエントリ削除部134、ファイルとディレクトリエントリの対応情報を管理するファイル/ディレクトリエントリ対応情報管理部135、ファイルの削除順序の問い合わせ及び設定を行う削除順序問い合わせ/設定部136、ファイルを削除する際に削除条件に合致しているか否かを判断する削除条件判断部137、ファイルの削除条件の問い合わせ及び設定を行う削除条件問い合わせ/設定部138、各ファイルを削除するときの条件に関する情報を管理する削除条件情報管理部139、及び削除候補ファイル問い合わせ部140を備えている。

【0030】上記削除順序情報管理部133は、上記削除順序情報格納部114にファイルの削除順序に関する情報を格納して、ファイルシステム110内のファイルの削除順序を管理している。

【0031】上記ファイル/ディレクトリエントリ対応

情報管理部135は、ファイルシステム110内のファイルとディレクトリエントリとの対応情報を、ファイル／ディレクトリエントリ対応情報格納部115に格納して、ファイルシステム110内のファイルとディレクトリエントリとの対応情報を管理している。

【0032】上記削除条件情報管理部139は、ファイルシステム110内の各ファイルの削除条件に関する情報を、削除条件情報格納部116に格納して、ファイルシステム110内の各ファイルの削除条件に関する情報を管理している。

【0033】次に、上記第一の実施例の本発明の要部に関する動作を説明する。

① ファイル作成時

残量把握機構131aは、常時、ファイルシステムの空き容量を監視している。そして、ファイル作成／削除機構131bが新たにファイルを作成するとき、ファイルシステムに空き容量が無くなると、そのことをファイル削除部132に通知する。ファイル削除部132は、削除順序情報管理部133に問い合わせ、削除対象となるファイルの情報を取得する。そして、ファイル削除部132は、この削除順序情報管理部133から通知されたファイルの削除を、ファイル作成／削除機構131bに依頼する。ファイル作成／削除機構131bは、この依頼を受け取ると、通知されたファイルを削除する。そして、新たなファイルを作成する。

【0034】② ファイル削除時

ファイル作成／削除機構131bは、ファイル削除の依頼を受け取ると、ファイル削除部132を介して削除順序情報管理部133から削除対象のファイルを通知してもらい、そのファイルを削除する。

【0035】③ ディレクトリからのファイルの削除
ファイルシステムがディレクトリ構造である場合、単にファイルのみを削除してしまうと、ディレクトリとの一貫性がとれなくなる。このため、あるファイルを削除する場合、そのファイルが属するディレクトリのエントリも一緒に削除する。

【0036】すなわち、ファイル削除部132は、削除対象のファイルの情報を得ると、そのファイルの削除をファイル作成／削除部131bに依頼すると同時に、ファイル／ディレクトリエントリ対応情報管理部135に上記削除対象のファイルが存在するディレクトリを問い合わせ、そのディレクトリの情報を得る。ファイル削除部132は、上記得たディレクトリのエントリからその削除対象となったファイルを削除するように、ディレクトリエントリ削除部134に依頼する。ディレクトリエントリ削除部134は、この依頼を受けてファイル／ディレクトリエントリ対応情報管理部135を介してファイル／ディレクトリエントリ対応情報格納部115に格納されている上記削除対象となったファイルと上記ディレクトリエントリとの対応関係を解除すると同時に、デ

ィレクトリ中のファイルのエントリを削除する。

【0037】④ ファイルの削除順序の問い合わせと設定

例えば、ファイルの削除順序が作成日時の順序になっていたものを、更新順序順に変更する。また、現在、ファイルの削除順序がどのようになっているのかを問い合わせ可能にする。

【0038】上記機能は、削除順序問い合わせ／設定部136により実行される。削除順序合わせ／設定部136は、削除順序情報管理部133に問い合わせ、ファイルの削除順序の情報を得たり、削除順序情報管理部133に対してファイルの削除順序の変更を依頼する。

【0039】⑤ ファイルの削除が条件に合致しているかの判断

例えば、配下にファイルが残っているディレクトリは削除しない、削除は作成日時順に行うが、アクセスが例えば3日以内にあったものは例外として削除しない、または作成日時が3日以内のものは削除しないなどの条件に基づいて、ファイルを削除する場合などに用いる。

【0040】上記①～③において、ファイル削除部132が削除対象のファイルを見つけた場合、そのファイルを削除してもよいか否かの条件を削除条件情報管理部139に問い合わせる。そして、ファイル削除部132は、削除条件情報管理部139から受け取った条件を削除条件判断部137に通知して、上記ファイルを削除してもよいか否かを確認する。そして、削除条件判断部137から削除しても良いとの結果を受け取った場合のみ、上記ファイルの削除をファイル作成／削除機構131bに依頼する。

【0041】また、削除条件問い合わせ／設定部138は、削除条件情報管理部139にファイルの削除条件がどうなっているのかを問い合わせたり、さらには、新たなファイルの削除条件を設定する。

【0042】⑥ 削除候補となっているファイルの問い合わせ

例えば、実際にファイルシステム110に存在する最古のファイルを調べるインターフェースをアプリケーションプログラムに提供することにより、そのファイルシステム110にファイルを作成しているアプリケーションプログラムとの同期を取りやすくする。

【0043】例えば、インターネットにおいて電子掲示板を提供するアプリケーションプログラムのヒストリ(history)機能では10日前までの記事を保存していることになっていても、ファイルの空き容量が不足したので、ファイルシステム110の都合により7日以上前の古いファイルは全て消したというような場合に、この問い合わせ機能を使って、上記アプリケーションプログラムがファイルシステム110に対していつまでのファイルが残っているのかを確認し、それに合わせて上記ヒストリの内容を調整する。

【0044】削除順序問い合わせ/設定部136は、削除順序情報管理部133を介して削除順序情報格納部114に登録されている削除対象のファイルについての情報を得ることができる。そして、この情報をアプリケーションプログラムからの依頼に応じて、アプリケーションプログラムに通知する。

【0045】図2は、図1の基本構成の第一の実施例を示すブロック図である。ファイルシステム110は、ファイル情報を管理するファイル管理テーブル112及びファイルシステム110を管理するための情報が格納されているファイルシステム管理ブロック113等からなる。ファイルシステム110においては、ファイルの実体は、不図示のデータ領域と呼ばれる複数のデータブロックから成る領域に格納される。また、ファイル管理テーブル112は、#1～#Nのエントリ番号が付与されたN個のエントリを有し、これらの各エントリには各ファイル毎に作成されるファイル管理情報ブロックが格納される。このファイル管理情報ブロックには、ファイルの種類（普通のファイル、ディレクトリファイルなど）、そのファイルの実体が格納されているデータ領域上の場所に関する情報などが格納される。

【0046】ファイルシステム管理ブロック113は、フリーブロックカウンタ113a、先頭ポインタ113b、及び最終ポインタ113cを備えている。フリーブロックカウンタ113aは、上記データ領域内の空きデータブロックの数を計数するカウンタである。先頭ポインタ113bは、ファイル管理テーブル112中の最古に作成されたファイルのエントリ番号を指すポインタである。また、最終ポインタ113cは、ファイル管理テーブル112中の最新に作成されたファイルのエントリ番号を指すポインタである。

【0047】ファイルシステム管理機構130は、残量把握機構131a、ファイル作成/削除機構131b、及びファイル削除順序管理部131cを備えている。残量把握機構131aは、ファイルシステム110の空き容量を、常時、監視している。

【0048】ファイル作成/削除機構131bは、ファイルシステム110内のファイルの作成及び削除を行う。ファイル順序管理部131cは、ファイルの削除順序を管理している。

【0049】この図2のシステムでは、ファイル管理テーブル112にファイル情報のエントリをファイルの作成順に格納することで、ファイルの削除順序の管理を兼ねている。したがって、ファイルの削除はファイルの作成順に行うことを前提にしている。このため、ファイルの作成順とは別の順番でファイルを削除することはできない。さらに、いったんファイルを作成した後で、そのファイルの内容を更新してもファイルの削除順序を変更することは出来ない。

【0050】また、ファイルシステム110の初期状態

では、先頭ポインタ113b及び最終ポインタ113cは共に、どのエントリも指していない（すなわち、“0”を指す状態に設定される（図5（a）参照）。

【0051】図3は、上記図2のシステムにおけるファイル作成時のファイル順序管理部131cの動作を説明するフローチャートである。まず、最初に第1のファイルを作成する場合の動作から説明していく。ファイル順序管理部131cは、まず、最終ポインタ113cの値が“0”であるか否かを判別する（S101）。そして、“0”であれば、最終ポインタ113cに“1”を設定する（S104）。

【0052】続いて、最終ポインタ113cと先頭ポインタ113bの値が等しいか否かを判別する（S105）。両ポインタの値が等しい場合は、ファイル管理テーブル112が満杯になり、新たにファイルを作成することが不可能になったことを示す。

【0053】第一のファイルを作成する場合、まだ、両ポインタの値は等しくないので、最終ポインタ113cの示すファイル管理テーブル112のエントリ、すなわち、“1”のエントリにファイルを作成する（S107）。

【0054】そして、次に、上記作成したファイルに対するデータの書き込みを終了したと判別するまで（S108、YES）、フリーブロックカウンタ113aの値が“0”でないことを確認しながら（S109、YES）、上記ファイルのデータをデータ領域に書き込む（S111）。そして、データの書き込みが全て終了すると（S108、YES）、先頭ポインタ113bの値が“0”か否かを判別し（S112）、先頭ポインタ113bの値が“0”であれば、この先頭ポインタ113bに最終ポインタ113cの値を代入する（S113）。

【0055】以上のようにして、ファイルシステム110に最初のファイルが作成される。このとき、図5（b）に示すように、先頭ポインタ113bと最終ポインタ113cの値は、共に、“1”に設定される。

【0056】次に、第2のファイルを作成する場合には、まず、最終ポインタ113cの値が“0”であるか否かを判別する（S101）。このとき、最終ポインタ113cの値は“1”となっているので、最終ポインタ113cの値は“0”でないと判別して（S101、NO）、最終ポインタ113cの値を“1”増加する（S102）。

【0057】これにより、最終ポインタ113cの値は“2”になる。次に、最終ポインタ113cの値がファイル管理テーブル112の最終エントリよりも大きいか否かを判別する（S103）。この場合、まだ、小さいので（S103、NO）、次に、上記ステップS105で最終ポインタ113cと先頭ポインタ113bの値が等しいか否かを判別する（S105）。この場合も、

両ポインタの値は等しくないので(S105、NO)、最終ポインタ113cの示すファイル管理テーブル112の"2"のエントリにファイルを作成する(S107)。

【0058】以下、上記ステップS108→S109→S111を繰り返しながら第2のファイルのデータをデータ領域に書き込む。そして、全てのデータの書き込みが終了すると、先頭ポインタ113bが"0"か否かを判別し(S112)、"0"でないので(S112、NO)、処理を終了する。

【0059】この結果、図5(c)に示すように先頭ポインタ113bは"1"、最終ポインタ113cは"2"に設定される。この場合、ファイル管理テーブル112のエントリが合計9個であるとする、データ領域の空き容量が第9のファイルを作成するまで満杯にならないければ、上記第2のファイルの作成と同様な動作が第9のファイルを作成まで繰り返される。この間、最終ポインタ113cの値は、順次、"1"ずつ増加し、図5(d)に示すように"9"となる。この間、先頭ポインタ113bの値は"1"のままである。

【0060】次に、第10のファイルが作成されると、上記ステップS102で最終ポインタ113cの値が"10"に設定され、次のステップS103で最終ポインタ113cの値がファイル管理テーブル112の最終エントリである"9"よりも大きいと判別される(S103、YES)。そして、続くステップS104で最終ポインタ113cに"1"が代入される。このことにより、ステップS105で最終ポインタ113cと先頭ポインタ113bの値が共に"1"で等しいと判別される(S105、YES)、後述する図4のフローチャートで詳細に説明する処理により、先頭ポインタ113bの指すファイル管理テーブル112のエントリ"1"に作成された第1のファイルの削除処理が行われる(S106)。この結果、先頭ポインタ113bの値は"1"増加されて"2"となる。

【0061】そして、続いて、最終ポインタ113cの示すファイル管理テーブル112のエントリ"1"に第10のファイルが作成される(S107)。そして、上記ステップS108→S109→S111を繰り返しながら第10のファイルのデータをデータ領域に書き込む。そして、全てのデータの書き込みが終了すると、先頭ポインタ113bが"0"か否かを判別し(S112)、"0"でないので(S112、NO)、処理を終了する。

【0062】この結果、第10のファイルの作成が完了すると、図5(e)に示すように先頭ポインタ113bの値は"2"に、最終ポインタ113cの値は"1"に設定される。

【0063】次に、第11のファイルの作成時には、ステップS102で最終ポインタ113cが"1"増加さ

れて"2"となり、ステップ103でNOと判別された後、ステップS105で最終ポインタ113cと先頭ポインタ113bの値が共に"2"で等しいと判別される(S105、YES)。そして、ステップS106で先頭ポインタ113bの示すファイル管理テーブル112のエントリ"2"のファイルの削除処理が行われる。このとき、先頭ポインタ113bの値が"1"増加されて"3"となる。

【0064】続いて、最終ポインタ113cの示すファイル管理テーブル112の"2"のエントリに第11のファイルが作成される(S107)。そして、次に、ステップS108以降で第11のファイルのデータの書き込みが行われるが、データの書き込み途中でフリーブロックカウンタ113aの値が"0"となり、データ領域に空きのデータブロックが無くなると(S109、YES)、上記ステップS106と同様な先頭ポインタ113bの示すファイル管理テーブル112の"3"のエントリのファイルの削除処理が行われる(S110)。そして、上記ステップS108～S111を繰り返しながら第11のファイルのデータをデータ領域に書き込む。そして、全てのデータの書き込みが終了すると、先頭ポインタ113bが"0"か否かを判別し(S112)、"0"でないので(S112、NO)、処理を終了する。この結果、先頭ポインタ113bの値は、"1"増加して"4"となる。

【0065】次に、図4のファイル順序管理部131cの実行するファイルの削除処理の動作を説明する。まず、先頭ポインタ113bの値が"0"であるか否かを判別し(S121)、"0"であれば直ちに処理を終了する。先頭ポインタ113bの値が"0"であるということは、ファイルシステム110内に削除すべきファイルが無いことを意味する。

【0066】一方、上記ステップS121で先頭ポインタ113bの値が"0"で無ければ、先頭ポインタ113bの示すファイル管理テーブル112のエントリのファイルを削除する(S122)。

【0067】続いて、先頭ポインタ113bの値を"1"増加した後(S123)、先頭ポインタ113bの値がファイル管理テーブル112の最終エントリの値よりも大きいかな否かを判別する(S124)。そして、先頭ポインタ113bの値が最終エントリの値よりも大きくなければ(S124、NO)、次に、先頭ポインタ113bの示すファイル管理テーブル112のエントリが空きであるか否かを判別する(S126)。

【0068】そして、先頭ポインタ113bの示すファイル管理テーブル112のエントリが空きでなければ(S126、NO)、次に、最終ポインタ113cの示すファイル管理テーブル112のエントリが空きであるか否かを判別する(S128)。このとき、上記エントリが空きで無ければ(S128、NO)、処理を終了す

る。

【0069】以上の処理は、上述した第10のファイルを作成する場合に先頭ポインタ113bの示すファイル管理テーブル112のエントリの"1"のファイルを削除する場合に対応している。この処理は、ファイル管理テーブル112のエントリが満杯になった後、順次、ファイルを1個ずつ作成する場合に対応している。すなわち、この場合には、順次、最古のファイルから削除されていく。

【0070】また、上記ステップS124で先頭ポインタ113bの値がファイル管理テーブル112の最終エントリの値よりも大きい場合には(S124, YES), 先頭ポインタ113bに"1"を代入して(S125)、ステップS126に移る。

【0071】この処理は、今回、ステップS122でファイル管理テーブル112の最終エントリのファイルを削除した場合に対応している。また、先頭ポインタ113bの示すファイル管理テーブル112のエントリが空きである場合には(S126, YES), 先頭ポインタ113bに"0"を代入する(S127)。そして、また、上記ステップS128で最終ポインタ113cの示すファイル管理テーブル112のエントリが空きである場合には(S128, YES), 最終ポインタ113cに"0"を代入する(S129)。

【0072】以上の処理は、ファイル管理テーブル112のエントリが全て空きになった場合に対応する。すなわち、ファイルシステム110のファイルの使用量が"0"となり、ファイルシステム110が初期状態に戻ったことを意味している。

【0073】このファイルの削除処理の動作の一例を、図5(e)~(i)を参照しながら説明する。図5(e)に示すように第10のファイルが作成された状態にあるときに、第2のファイルから第10のファイルを、順次、削除する場合、まず、図4のフローチャートの処理により、先頭ポインタ113bの示すファイル管理テーブル112の"2"のエントリに作成されている第2のファイルがステップS122で削除され、先頭ポインタ113bの値がステップS123で"3"に設定される(図5(f)参照)。

【0074】続いて、先頭ポインタ113bの示すファイル管理テーブル112の"3"のエントリに作成されている第3のファイルがステップS122で削除され、先頭ポインタ113bの値がステップS123で"4"に設定される(図5(g)参照)。

【0075】以後、同様にして、第4~第9のファイルが順次削除され、先頭ポインタ113bと最終ポインタ113cの値は、共に、"1"に設定される(図5(h)参照)。このとき、先頭ポインタ113bの値は、図4のフローチャートのステップ125によって"1"に設定される。

【0076】続いて、ファイル管理テーブル112の"1"のエントリに作成されている第10のファイルが図4のフローチャートのステップS122で削除される。この後、ステップS126とステップS128で、それぞれ、先頭ポインタ113bと最終ポインタ113cの示すファイル管理テーブル112のエントリが空きであると判別され、ステップS127とS129で、それぞれ、先頭ポインタ113bと最終ポインタ113cに"0"が代入される(図5(i)参照)。

【0077】次に、図6は、上記第一の基本構成の第二の実施例を示すブロック図である。同図において、図2と同様のブロックには同じ符号を付与している。ファイル管理テーブル112'の各エントリのファイル管理情報ブロックは、ファイル管理テーブル112'の次のエントリへのポインタを示す次エントリポインタ1121を備えている。

【0078】ファイルシステム管理ブロック113'は、フリーブロックカウンタ113a、使用済みエントリ先頭ポインタ113b'、使用済みエントリ最終ポインタ113c'に加え、未使用エントリ先頭ポインタ113d及び未使用エントリ最終ポインタ113eを備えている。

【0079】使用済みエントリ先頭ポインタ113b'はファイル管理テーブル112'中の最古のファイルのエントリを指すポインタ、使用済みエントリ最終ポインタ113c'はファイル管理テーブル112'中の最新のファイルのエントリを指すポインタである。

【0080】また、未使用エントリ先頭ポインタ113dはファイル管理テーブル112'の未使用の先頭のエントリを指すポインタ、未使用エントリ最終ポインタ113eはファイル管理テーブル112'の未使用の最終のエントリを指すポインタである。

【0081】ファイルシステム管理機構130'は、残量把握機構131a、ファイル作成/削除機構131b'及びファイル順序管理部131c'を備えている。この図6に示すシステムにおいては、ファイル管理テーブル112'内の各ファイル情報管理ブロック内に次エントリポインタ1121を含ませることによって、ファイル管理テーブル112'内でのファイルの削除の順序情報をリストにより管理している。また、未使用エントリ先頭ポインタ113dと未使用エントリ最終ポインタ113eとを次エントリポインタ1121によりリストで連結することにより、ファイル管理テーブル112'の未使用のエントリもリストにより管理している。

【0082】このような構成のため、ファイル管理テーブル112'のファイルの削除順序を簡単に入れ換えることが可能になっている。例えば、あるファイルを更新した場合に、このファイルの削除順序を一番最後に移動することも可能である。この結果、この実施例では更新日時の古い順にファイルを削除することも可能になって

いる。また、削除順序とは無関係に、削除順序の途中にあるファイルも削除することも許可している。

【0083】以下、図7～図16を参照しながら、ファイル順序管理部131c'によって実行される、この実施例におけるファイルの作成及び削除の動作を説明する。初期状態では、図14(a)に示すように、使用済みエントリ先頭ポインタ113b'と使用済みエントリ最終ポインタ113c'は、共に"0"を指している。また、未使用エントリ先頭ポインタ113dはファイル管理テーブル112'の"1"のエントリを、未使用エントリ最終ポインタ113eはファイル管理テーブル112'の最終エントリ(この場合、"10")を指している。また、未使用エントリ先頭ポインタ113dの指すファイル管理テーブル112'の先頭のエントリ(この場合、"1")のファイル管理情報ブロックから未使用エントリ最終ポインタ113eの指すファイル管理テーブル112'のエントリ(この場合、"10")のファイル管理情報ブロックは、それぞれの次エントリポインタ1121によってリンクされており、未使用エントリ最終ポインタ113eの指すファイル管理テーブル112'のエントリ(この場合、"10")のファイル管理情報ブロックの次エントリポインタ1121には"0"が設定されている。

【0084】図7は、ファイル順序管理部131c'によるファイル作成処理を説明するフローチャートである。まず、未使用エントリ先頭ポインタ113dの値が"0"であるか否かを判別する(S201)。最初に第1のファイルを作成する時には、未使用エントリ先頭ポインタ113dの値は"1"なので、次に、未使用エントリ先頭ポインタ113dの値を変数xに代入する(S203)。この結果、変数xの値は"1"になる。

【0085】続いて、変数xの示すファイル管理テーブル112'のエントリの次エントリポインタ1121の値を、未使用エントリ先頭ポインタ113dに代入する(S204)。この結果、図14(b)に示すように未使用エントリ先頭ポインタ113dの値は"2"に設定される。

【0086】次に、未使用エントリ先頭ポインタ113dが"0"であるか否かを判別する(S205)。この場合、"0"でないので、続いて、使用済みエントリ最終ポインタ113c'の値が"0"であるか否かを判別する(S207)。この場合、"0"なので、続いて、使用済みエントリ先頭ポインタ113b'と使用済みエントリ最終ポインタ113c'に共に変数xの値である"1"を設定する(S208)。

【0087】この結果、図14(b)に示すように、使用済みエントリ先頭ポインタ113b'と使用済みエントリ最終ポインタ113c'が共に、ファイル管理テーブル112'の"1"のエントリを指すようになる。

【0088】次に、変数xの示すファイル管理テーブル

112'のエントリの次エントリポインタ1121に"0"を代入する(S210)。この結果、図14(b)に示すようにファイル管理テーブル112'の"1"のエントリの次エントリポインタ1121に"0"が設定される。そして、変数xの示すファイル管理テーブル112'のエントリに第1のファイルを作成する(S211)。

【0089】続いて、第1のファイルのデータの作成がファイルシステム110のデータ領域のデータブロックに書き込まれる(S212～S215)。この動作は、図3のフローチャートのステップS108～S111の処理と同様である。すなわち、ステップS213でフリーブロックカウンタ113aの値が"0"であると判別されるまで、ステップS212→S213→S215が繰り返され、ステップS212でデータの書き込みが終了したと判別すると、処理を終了する。

【0090】このようにして、まず、ファイル管理テーブル112'の"1"のエントリに第1のファイルが作成される。この結果、使用済みエントリ先頭ポインタ113b'と使用済みエントリ最終ポインタ113c'は共にファイル管理テーブル112'の"1"のエントリを指す。また、この"1"のエントリの次エントリポインタ1121は"0"に設定される。さらに、未使用エントリ先頭ポインタ113dがファイル管理テーブル112'の"2"のエントリを指す。そして、未使用エントリ最終ポインタ113dは、初期状態のときと変わらずファイル管理テーブル112'の"10"のエントリを指したままである(図14(b)参照)。

【0091】次に、第2のファイルを作成する場合、上記第1のファイルの作成時と同様な動作が行われ、ステップS203で変数xに未使用エントリ先頭ポインタ113dの値"2"が設定される。そして、ステップS205で未使用エントリ先頭ポインタ113dの値が"0"でないかと判別された後、次のステップS207で使用済みエントリ最終ポインタ113c'の値が"0"でないかと判別される。

【0092】そして、次に、ステップS209で使用済みエントリ最終ポインタ113c'の示すファイル管理テーブル112'の"1"のエントリの次エントリポインタ1121に変数xの値を代入した後、使用済みエントリ最終ポインタ113c'に変数xの値を代入する。

【0093】この結果、ファイル管理テーブル112'の"1"のエントリ"の次エントリポインタ1121の値は"2"となる。また、使用済みエントリ最終ポインタ113c'の値は"2"となる。

【0094】次に、変数xの示すファイル管理テーブル112'の"2"のエントリの次エントリポインタ1121に"0"が代入される(S210)。続いて、変数xの示すファイル管理テーブル112'の"2"のエントリに第2のファイルが作成される(S211)。そし

て、続くステップS212～S215の処理により第2のファイルのデータがファイルシステム110のデータ領域に書き込まれる。

【0095】以上のようにして、第2のファイルが作成され、図14(c)に示すように、使用済みエントリ最終ポインタ113c'がファイル管理テーブル112'の"2"のエントリを指すようになる。また、未使用エントリ先頭ポインタ113dがファイル管理テーブル112'の"3"のエントリを指すようになる。さらに、ファイル管理テーブル112'の"1"のエントリの次エントリポインタ1121がファイル管理テーブル112'の"2"のエントリを指し、ファイル管理テーブル112'の"2"のエントリの次エントリポインタ1121が"0"に設定される。

【0096】この結果、使用済みエントリ先頭ポインタ113b'と使用済みエントリ最終ポインタ113c'とファイル管理テーブル112'の"1"と"2"のエントリの次エントリポインタ1121によって、ファイルの削除順序のリストが作成される。また、未使用エントリ先頭ポインタ113dと未使用エントリ最終ポインタ113e及びファイル管理テーブル112'の"3"～"10"のエントリによって、未使用のファイルのリストが作成される。

【0097】以後、同様に、第3～第10のファイルが作成されると、ファイル管理テーブル112'は図15(d)に示すようになる。すなわち、第9のファイルの作成を終了した時点で、未使用エントリ最終ポインタ113eの値は"10"に設定される。また、使用済みエントリ最終ポインタ113c'の値は"9"に設定される。そして、ファイル管理テーブル112'の"9"と"10"のエントリの次エントリポインタの値が共に"0"に設定される。

【0098】ファイル管理テーブル112'が以上の状態にあるときに、第10のファイルの作成が開始される。そして、ステップS203で変数xに未使用エントリ先頭ポインタ113dの値が代入され、変数xの値は"10"になる。続いて、ステップS204で変数xの示すファイル管理テーブル112'のエントリ"10"の次エントリポインタ1121の値が未使用エントリ先頭ポインタ113dに代入され、未使用エントリ先頭ポインタ113dの値は"0"に設定される。

【0099】そして、次に、ステップS205で未使用エントリ先頭ポインタ113dの値が"0"に等しいと判別され、ステップS206で未使用エントリ最終ポインタ113eに"0"が設定される。次に、ステップS207で使用済みエントリ最終ポインタ113c'の値が"0"でないと判別された後、使用済みエントリ最終ポインタ113c'の示すエントリの次エントリポインタ1121に変数xの値が代入され、使用済みエントリ最終ポインタ113c'の値は"10"となる。

【0100】続いて、変数xの示すファイル管理テーブル112'のエントリ"10"の次エントリポインタ1121に"0"が設定される(S210)。そして、変数xの示すファイル管理テーブル112'のエントリ"10"に第10のファイルが作成される(S211)。続いて、ステップS212～S215の処理によって、ファイルシステム110のデータ領域に第10のファイルのデータが書き込まれる。

【0101】以上の結果、図15(d)に示すように、使用済みエントリ先頭ポインタ113b'と使用済みエントリ最終ポインタ113c'が、それぞれ、ファイル管理テーブル112'の"1"のエントリと"10"のエントリを指すようになる。また、未使用エントリ先頭ポインタ113dと未使用エントリ最終ポインタ113eが共に"0"となる。このようにして、使用済みエントリ先頭ポインタ113b'と使用済みエントリ最終ポインタ113c'及びファイル管理テーブル112'の"1"～"10"のエントリの次エントリポインタ1121とによって、ファイル管理テーブル112'には第1～第10のファイルのリストが作成され、ファイル管理テーブル112'の空きエントリは"0"となる。これは、未使用エントリ先頭ポインタ113dの値が"0"となっていることによって示される。

【0102】この状態のときに、第11のファイルの作成を行うと、最初のステップS201で未使用エントリ先頭ポインタ113dの値が"0"であると判別され、ステップS202で使用済みエントリ先頭ポインタ113b'の示すファイル管理テーブル112'のエントリに作成されたファイルの削除処理が行われる。

【0103】図9はこのファイルの削除処理を説明するフローチャートである。まず、使用済みエントリ先頭ポインタ113b'の値が"0"であるか否かを判別する(S221)。そして、"0"でなければ、使用済みエントリ先頭ポインタ113b'の指すファイル管理テーブル112'のエントリに作成されているファイルを削除する(S222)。

【0104】これにより、ファイル管理テーブル112'のエントリ"1"に作成されている第1のファイルが削除される。次に、使用済みエントリ先頭ポインタ113b'の値を変数xに代入する(S223)。

【0105】これにより、変数xの値が"1"になる。続いて、変数xの示すファイル管理テーブル112'のエントリの次エントリポインタ1121の値が"0"であるか否かを判別する(S224)。

【0106】この場合、ファイル管理テーブル112'のエントリ"1"の次エントリポインタ1121は図15(d)に示すように"0"でないため、使用済みエントリ先頭ポインタ113b'に変数xの示すファイル管理テーブル112'のエントリの次エントリポインタ1121の値を代入する(S225)。

【0107】これにより、使用済みエントリ先頭ポイント113b'の値が"2"に設定される。次に、未使用エントリ最終ポイント113dの値が"0"であるか否かを判別する(S228)。

【0108】この場合、未使用エントリ最終ポイント113dの値は"0"であるので、未使用エントリ先頭ポイント113dに変数xの値を代入する(S229)。これにより、未使用エントリ先頭ポイント113dに"1"が設定される。

【0109】次に、未使用エントリ最終ポイント113dに変数xの値を代入する(S231)。これにより、未使用エントリ最終ポイント113dに"1"が設定される。

【0110】そして、変数xの示すファイル管理テーブル112'のエントリの次エントリポイント1121に"0"を代入する(S232)。これにより、ファイル管理テーブル112'のエントリ"1"の次エントリポイント1121に"0"が設定される。

【0111】以上のようにして、第1のファイルが削除され、そして、図15(e)に示すように、未使用エントリ先頭ポイント113dと未使用エントリ最終ポイント113eは、共に、ファイル管理テーブル112'のエントリ"1"を指す。また、ファイル管理テーブル112'のエントリ"1"の次エントリポイント1121が"0"に設定される。また、使用済みエントリ先頭ポイント113b'は、ファイル管理テーブル112'のエントリ"2"を指す。

【0112】再び、図7のフローチャートの説明に戻る。上記のようにして、ステップS202のファイル削除処理が終了した後、ステップS203で未使用エントリ先頭ポイント113dの値を変数xに代入する。

【0113】これにより、変数xの値は"1"になる。次に、ステップS204で変数xの示すファイル管理テーブル112'のエントリの次エントリポイント1121の値を、未使用エントリ先頭ポイント113dに代入する。

【0114】これにより、未使用エントリ先頭ポイント113dには"0"が設定される。次に、ステップS205で未使用エントリ先頭ポイント113dが"0"であると判別され、未使用エントリ最終ポイント113eに"0"が設定される(S206)。

【0115】そして、ステップS207で、使用済みエントリ最終ポイント113c'の値が"0"でないと判別され、ステップS209で使用済みエントリ最終ポイント113c'の示すファイル管理テーブル112'のエントリの次エントリポイント1121に変数xの値を代入した後、使用済みエントリ最終ポイント113c'に変数xの値を代入する。

【0116】これにより、ファイル管理テーブル112'の"10"のエントリの次エントリポイント112

1は、"1"に設定され、また、使用済みエントリ最終ポイント113c'の値は"1"に設定される。

【0117】続いて、変数xの示すファイル管理テーブル112'のエントリの次エントリポイント1121に"0"を代入する(S210)。これにより、ファイル管理テーブル112'のエントリ"1"の次エントリポイント1121は、"0"に設定される。

【0118】続いて、変数xの示すファイル管理テーブル112'のエントリに第11のファイルを作成する(S211)。これにより、ファイル管理テーブル112'のエントリ"1"に第11のファイルが作成される。

【0119】そして、図8のフローチャートのステップS212~S215の処理により、第11のファイルのデータをファイルシステム110のデータ領域に書き込む。この結果、未使用エントリ先頭ポイント113dと未使用エントリ最終ポイント113eは、共に、"0"を示す。また、使用済みエントリ先頭ポイント113b'と使用済みエントリ最終ポイント113c'は、共に、ファイル管理テーブル112'の"1"のエントリを指すようになる。

【0120】また、この後、新たにファイルを作成せずに、ファイルを連続して削除していくと、ファイル管理テーブル112'に作成されている最後のファイルを削除するとき、図9のフローチャートのステップS224で、変数xの示すファイル管理テーブル112'のエントリの次エントリポイント1121の値が"0"であると判別される。そして、ステップS206とS207で、使用済みエントリ先頭ポイント113b'と使用済みエントリ最終ポイント113c'に"0"が設定される。この結果、ファイル管理テーブル112'の使用量は"0"となる。

【0121】上記ファイルの削除処理は、使用済みエントリ先頭ポイント113b'の指すファイル管理テーブル112'のエントリに作成されているファイルを削除するものであった。すなわち、使用済みエントリ先頭ポイント113b'と使用済みエントリ最終ポイント113c'及び次エントリポイント1121によって作成されるリストの先頭のファイルからそのリスト順にファイルを削除していくものであった。

【0122】次に、使用済みエントリ先頭ポイント113b'と使用済みエントリ最終ポイント113c'及び次エントリポイント1121によって構成されるリストの中間に位置するファイルを削除する方法を説明する。

【0123】図15(e)の状態にあるときに、ファイル管理テーブル112'のエントリ"5"のファイルを削除する場合の動作を、図10~図11のフローチャートを参照しながら説明する。

【0124】まず、使用済みエントリ先頭ポイント113b'の値を変数xに代入する(S241)。これによ

り、変数xには"2"の値が代入される。

【0125】次に、変数xの値が"0"であるか否かを判別する(S242)。この場合、"0"でないので、次に、変数xの値が削除対象のファイルのエントリ位置に該当するか否かを判別する(S243)。

【0126】この場合、変数xの値が"5"でないで、変数xの値を変数oldxに代入する(S244)。そして、変数xの示すファイル管理テーブル112'のエントリの次エントリポインタ1121を変数xに代入する(S245)。

【0127】これにより、変数xの値は"3"に設定される。上記ステップS242～S245の処理が繰り返されて、変数xの値は、順次、"1"づつ増加し、やがて、"5"となる。このとき、変数oldxの値は"4"となる。

【0128】そして、ステップS243で変数xの値が削除対象のファイルのエントリ位置に等しいと判別され、変数xの示すファイル管理テーブル112'のエントリ"5"に作成されているファイルを削除する(S246)。

【0129】続いて、使用済みエントリ先頭ポインタ113b'の値が変数xの値に等しいか否かを判別する(S247)。この場合、使用済みエントリ先頭ポインタ113b'の値は"2"なので(NO)、次に、変数oldxの示すファイル管理テーブル112'のエントリの次エントリポインタ1121に、変数xの示すファイル管理テーブル112'のエントリの次エントリポインタ1121の値を設定する(S250)。

【0130】この結果、ファイル管理テーブル112'の"4"のエントリの次エントリポインタ1121が"6"となる。次に、変数xの示すエントリの次エントリポインタ1121の値が"0"であるか否かを判別する(S251)。この場合、変数xの示すエントリの次エントリポインタ1121の値は"6"であるので(NO)、続いて、未使用エントリ最終ポインタ113eの値が"0"であるか否かを判別する(S255)。

【0131】この場合、未使用エントリ最終ポインタ113eの値は"1"であるので(NO)、次に、未使用エントリ最終ポインタ113eの示すファイル管理テーブル112'の"1"のエントリの次エントリポインタ1121に、変数xの値を代入する(S256)。

【0132】これにより、ファイル管理テーブル112'の"1"のエントリの次エントリポインタ1121に"5"が設定される。続いて、未使用エントリ最終ポインタ113eに変数xの値を代入する(S258)。

【0133】この結果、未使用エントリ最終ポインタ113eには"5"が設定される。続いて、変数xの示すファイル管理テーブル112'のエントリの次エントリポインタ1121に"0"を設定する(S259)。

【0134】この結果、ファイル管理テーブル112'

の"5"のエントリの次エントリポインタ1121には"0"が設定される。以上の結果、図15(f)に示すように、使用済みエントリ先頭ポインタ113b'と使用済みエントリ最終ポインタ113eは、それぞれ、ファイル管理テーブル112'の"2"と"10"を指すようになる。また、未使用エントリ先頭ポインタ113dと未使用エントリ最終ポインタ113eは、ファイル管理テーブル112'の"1"と"5"のエントリを指すようになる。また、ファイル管理テーブル112'の"1"のエントリの次エントリポインタ1121は、"5"のエントリを指すようになり、"5"のエントリの次エントリポインタ1121は"0"に設定される。

【0135】次に、図15(f)に示す状態にあるときに、ファイル管理テーブル112'のエントリ"2"に作成されているファイルを削除する場合の動作を説明する。この場合、ステップS241で使用済みエントリ先頭ポインタ113b'の値"2"が変数xに代入される。次に、変数xの値が"0"でないかと判別された後、変数xの値が削除対象のファイルのエントリ位置に等しいか否かを判別する(S243)。

【0136】この場合、共に、値が等しいので、変数xの示すファイル管理テーブル112のエントリのファイルを削除し(S246)、次に、使用済みエントリ先頭ポインタ113b'の値が変数xの値と等しいか否かを判別する(S247)。この場合、共に、"2"であり、等しいので(YES)、変数xの示すファイル管理テーブル112'のエントリの次エントリポインタ1121の値が"0"であるか否かが判別される(S248)。

【0137】この場合、"0"ではないので、使用済みエントリ先頭ポインタ113b'に変数xの示すファイル管理テーブル112'のエントリの次エントリポインタ1121の値が代入される(S249)。

【0138】これにより、使用済みエントリ先頭ポインタ113b'の値が"3"に設定される。次に、ステップS255で未使用エントリ最終ポインタ113eの値が"0"でないと判別され(NO)、続いて、ステップS256で未使用エントリ最終ポインタ113dの示すファイル管理テーブル112'の"5"のエントリの次エントリポインタ1121に変数xの値が設定される。

【0139】この結果、ファイル管理テーブル112'の"5"のエントリの次エントリポインタ1121には"2"が設定される。続いて、ステップS258で未使用エントリ最終ポインタ113dに変数xの値が代入される。

【0140】これにより、未使用エントリ最終ポインタ113eの値が"2"になる。次に、ステップS259で変数xの示すファイル管理テーブル112'のエントリの次エントリポインタ1121に"0"が代入される。

【0141】これにより、ファイル管理テーブル112'のエントリ"2"の次エントリポインタ1121が"0"となる。この結果、ファイル管理テーブル112'には、ファイルの削除順序のリストとして、ファイル管理テーブル112'のエントリ"1"→"5"→"2"のリストが作成される。

【0142】次に、ファイル管理テーブル112'が図15(d)に示す状態にあるときに、ファイル管理テーブル112'の"10"のエントリに作成されたファイルを削除する場合の動作を説明する。

【0143】この場合、上記ステップS241～S246の処理が行われ、ファイル管理テーブル112'の"10"のエントリに作成されたファイルが削除される。このとき、変数xの値は"10"、変数oldxの値は"9"となっている。

【0144】続いて、ステップS247で使用済みエントリ先頭ポインタ113b'の値が変数xの値に等しくないと判別された後、ステップS250の処理によりファイル管理テーブル112'の"9"のエントリの次エントリポインタ1121に"0"が設定される。

【0145】次に、ステップS251で変数xの示す"10"のエントリの次エントリポインタ1121の値が"0"であると判別され(YES)、続いて、使用済みエントリ最終ポインタ113c'に変数oldxの示すファイル管理テーブル112'の"9"の値が設定される(S252)。

【0146】次に、ステップS255で未使用エントリ最終ポインタ113eの値が"0"であると判別され(YES)、続いて、未使用エントリ先頭ポインタ113dに変数xの値"10"が設定される。

【0147】続いて、ステップS258で未使用エントリ最終ポインタ113eに変数xの値"10"が設定された後、次のステップS259で変数xの示すファイル管理テーブル112'のエントリ"10"の次エントリポインタ1121に"0"が設定される。

【0148】次に、ファイル管理テーブル112'が図14(b)に示す状態にあるときに、ファイル管理テーブル112'のエントリ"1"に作成されたファイルを削除する場合の動作を説明する。

【0149】この場合、ステップS241→S242→S243→S246の処理が順次行われ、上記ファイル管理テーブル112'の"1"のエントリに作成されたファイルが削除される。この時、変数xには"1"が設定されている。

【0150】次に、ステップS247で使用済みエントリ先頭ポインタ113b'の値と変数xの値が等しいと判別され(YES)、次に、ステップS248で変数xの示すファイル管理テーブル112'の"1"のエントリの次エントリポインタ1121の値が"0"であると判別される(YES)。

【0151】そして、使用済みエントリ先頭ポインタ113b'に"0"が設定され(S253)、次に、使用済みエントリ最終ポインタ113c'に"0"が設定される(S254)。

【0152】そして、次に、ステップS255で未使用エントリ最終ポインタ113eの値が"0"でないと判別され(NO)、続いて、ステップS256で未使用エントリ最終ポインタ113eの示す"10"のエントリの次エントリポインタ1121に変数xの値"1"が設定される。

【0153】次に、ステップS258で未使用エントリ最終ポインタ113eに変数xの値"1"が設定された後、変数xの示すファイル管理テーブル112'の"1"のエントリの次エントリポインタ1121に"0"が設定される。

【0154】以上の結果、使用済みエントリ先頭ポインタ113b'と使用済みエントリ最終ポインタ113c'が共に"0"に設定される。また、未使用エントリ先頭ポインタ113dが"2"に未使用エントリ最終ポインタ113eが"1"に設定される。

【0155】また、ステップS242で変数xに代入された使用済みエントリ先頭ポインタ113b'の値が"0"と判別した場合は(YES)、ファイル管理テーブル112'には削除対象のファイルが存在しないので、直ちに、処理を終了する。次に、図12～図13は、更新したファイルを削除対象の一番最後に設定するファイル順序管理部131cの動作を説明するフローチャートである。

【0156】以下、ファイル管理テーブル112'が図15(f)の状態にあるときに、ファイル管理テーブル112'の"3"のエントリに作成されたファイルを更新する場合を例に取り上げて説明する。この場合も、削除順序リストに従うことなく、任意のファイルを更新することが可能である。

【0157】図12のフローチャートにおいて、ステップS271～S275の処理は、図10のフローチャートのステップS241～S245の処理と同様である。上記ファイル管理テーブル112'の"3"のエントリに作成されたファイルを更新する場合、変数xに"3"が、変数oldxに"2"が設定された後、ステップS276の処理に移る。

【0158】次に、変数xの値が使用済みエントリ先頭ポインタ113b'の値と等しいか否かが判別され(S276)、この場合、等しくないで(NO)、変数oldxの示すファイル管理テーブル112'のエントリの次エントリポインタ1121に、変数xの示すファイル管理テーブル112'のエントリの次エントリポインタ1121の値が設定される(S279)。

【0159】これにより、ファイル管理テーブル112'の"2"のエントリの次エントリポインタ1121

に"4"が設定される。続いて、変数xの示すファイル管理テーブル112'のエントリの次エントリポインタ1121の値が"0"であるか否かが判別される(S280)。

【0160】この場合、"0"でないので(NO)、次に、使用済みエントリ最終ポインタ113c'の示すエントリの次エントリポインタ1121に変数xの値を代入し、続いて、使用済みエントリ先頭ポインタ113b'の示すエントリの次エントリポインタ1121に変数xの値を設定し、使用済みエントリ最終ポインタ113c'に変数xの値を代入する(S282)。

【0161】これにより、ファイル管理テーブル112'のエントリ"10"の次エントリポインタ1121に"3"が設定される。また、使用済みエントリ最終ポインタ113c'に"3"が設定される。

【0162】続いて、変数xの示すファイル管理テーブル112'のエントリの次エントリポインタ1121に"0"を設定する(S283)。これにより、ファイル管理テーブル112'のエントリ"3"の次エントリポインタ1121に"0"が設定される。

【0163】次に、変数xの示すファイル管理テーブル112'のエントリに作成されたファイルの更新処理を行う(S284)。続いて、上記ファイルのデータの書き込み処理を行う(S285~S288)。このステップS285~S288の処理は、図8のステップS212~S215の処理と同様である。すなわち、この場合においても、データの書き込み途中でファイルシステム110のデータ領域に空きのデータブロックが無くなった場合には、図9のフローチャートに従って使用済みエントリ先頭ポインタ113b'の示すファイル管理テーブル112'のエントリに作成されたファイルを削除する。

【0164】以上の結果、図16(g)に示すように、使用済みエントリ先頭ポインタ113b'は"2"に、使用済みエントリ最終ポインタ113c'は"3"に、未使用エントリ先頭ポインタ113dは"1"に、未使用エントリ最終ポインタ113eは"5"に設定される。また、ファイル管理テーブル112'の各エントリの次エントリポインタ1121の内容は、図16(g)に示すようになる。

【0165】次に、ファイル管理テーブル112'が図15(f)の状態にあるときに、ファイル管理テーブル112'のエントリ"10"に作成されたファイルを更新する場合の動作を説明する。

【0166】この場合には、ステップS279まで、変数xの値が"10"、変数oldxの値が"9"に対して上記と同様の処理が行われた後、ステップS280で変数xの示すファイル管理テーブル112'のエントリ"10"の次エントリポインタ1121の値が"0"であると判別される(YES)。

【0167】以後、ステップS284以降の処理により、ファイル管理テーブル112'の"10"のエントリに作成されたファイルのデータの更新がおこなわれる。この結果、図15(f)に示すように、使用済みエントリ最終ポインタ113c'が"10"に設定され、エントリ"10"の次エントリポインタ1121が"0"に設定される。

【0168】次に、ファイル管理テーブル112'の状態が図16(g)に示す状態にあるときに、ファイル管理テーブル112'の"2"のエントリに作成されたファイルを更新する場合の動作を説明する。

【0169】この場合には、ステップS276まで、変数xの値が"2"に対して上記と同様の処理が行われた後、変数xの示すファイル管理テーブル112'のエントリ"9"の次エントリポインタ1121の値が"0"であるか否かが判別される(S277)。

【0170】そして、この場合、"0"ではないので(NO)、使用済みエントリ先頭ポインタ113b'に変数xの示すファイル管理テーブル112'のエントリの次エントリポインタ1121の値が代入される。

【0171】これにより、使用済みエントリ先頭ポインタ113b'の値は"4"となる。次に、ステップS282で、使用済みエントリ最終ポインタ113c'の示すファイル管理テーブル112'のエントリの次エントリポインタ1121に変数xの値が設定された後、使用済みエントリ最終ポインタ113c'に変数xの値が設定される。

【0172】これにより、ファイル管理テーブル112'のエントリ"3"の次エントリポインタ1121が"2"となる。また、使用済みエントリ最終ポインタ113c'が"2"になる。

【0173】続いて、ステップS283で変数xの示すファイル管理テーブル112'のエントリの次エントリポインタ1121に"0"が設定される。これにより、ファイル管理テーブル112'の"2"のエントリの次エントリポインタ1121が"0"となる。

【0174】以後、ステップS284以降の処理により、ファイル管理テーブル112'の"2"のエントリに作成されたファイルのデータの更新がおこなわれる。この結果、データが更新されたファイル管理テーブル112'の"2"のエントリに作成されたファイルが、ファイル削除順序リストの最後に位置するようになる。

【0175】このように、図12~図13のフローチャートに示す処理では、使用済みエントリ最終ポインタ113c'と次エントリポインタ1121の操作により、更新されたファイルが削除順序リストの最後につながれるように、ファイル管理テーブル112'の内容を更新するので、ファイルを更新日時(更新していない場合には作成日時を更新日時とみなす)の古い順から削除することができる。次に、図17は、上記第一の基本構成の

第三の実施例を示すブロック図である。同図において、図6と同様のブロックには同じ符号を付与している。

【0176】この第三の実施例は、上記第二の実施例を基にしており、上記第二の実施例との相違点は、ファイル管理テーブル112”の各エントリに、そのエントリが指すファイルが存在するディレクトリのファイル管理テーブル112”へのポインタであるディレクトリポインタ1123を設けていることと、ファイルシステム管理機構130”が、さらにディレクトリエントリ削除部131dを備えていることである。

【0177】この第三の実施例では、ファイルを作成、削除及び更新する際の動作は、上記第二の実施例の場合と同様であり、第二の実施例との相違点は以下の通りである。

【0178】① ファイル作成時に、ファイル管理テーブル112”内のディレクトリポインタ1123に、その作成するファイルが存在するディレクトリのファイル管理テーブル112”内のエントリへのポインタを書き込む。

【0179】② ファイル削除時に、その削除対象のファイルのファイル管理テーブル112”内のディレクトリポインタ1123を参照して、そのファイルが存在するディレクトリを見つけ出し、そのディレクトリから該当するエントリを削除する。

【0180】上記①の処理はファイル作成／削除機構131b”が、上記②の処理は、ディレクトリエントリ削除部134が実行する。図18は、ファイルシステム110のディレクトリ構成例を示す図である。同図に示す例では、ディレクトリAがルートディレクトリとなっており、この下にディレクトリB、Cの2つのディレクトリ（サブディレクトリ）とファイルA、Bの2つのファイルがある。また、上記ディレクトリBの下にはファイルC、Dの2つのファイルがあり、また、ディレクトリCの下にはファイルE、F、Gの3つのファイルがある。

【0181】次に、図19は、ファイルシステム110内のデータブロック内に作成される上記図18のディレクトリ構成に関する情報のデータ構造を示す図である。ディレクトリAは、図19（a）に示すように、自分自身、自己の上のディレクトリ、さらに、自己の下に存在するディレクトリB、ディレクトリC、ファイルA及びファイルBに関する位置情報を備えている。

【0182】ディレクトリBは、図19（b）に示すように、自分自身、自己の上のディレクトリ、さらに、自己の下に存在するファイルC及びファイルDに関する位置情報を備えている。

【0183】ディレクトリCは、図19（c）に示すように、自分自身、自己の上のディレクトリ、さらに、自己の下に存在するファイルE、ファイルF及びファイルGに関する位置情報を備えている。

【0184】図20は、上記図19に示すディレクトリのデータ構造を管理しているファイルシステム110内のファイル管理テーブル112”とファイルシステム管理ブロック113”の構成図である。

【0185】ディレクトリAに対するディレクトリポインタ1123は、ファイル管理テーブル112”のエントリ”1”のファイル管理情報ブロックに格納されている。このディレクトリポインタ1123の値は”0”である。また、このファイル管理情報ブロックの次エントリポインタ1121には”2”が設定されている。また、ディレクトリBに対するディレクトリポインタ1123は、ファイル管理テーブル112”のエントリ”2”のファイル管理情報ブロックに格納されている。このディレクトリポインタ1123の値は”1”である。また、このファイル管理情報ブロックの次エントリポインタ1121には”3”が設定されている。

【0186】また、ファイルCに対するディレクトリポインタ1123は、ファイル管理テーブル112”のエントリ”3”のファイル管理情報ブロックに格納されている。このディレクトリポインタ1123の値は”2”である。また、このファイル管理情報ブロックの次エントリポインタ1121には”4”が設定されている。

【0187】また、ファイルDに対するディレクトリポインタ1123は、ファイル管理テーブル112”のエントリ”4”のファイル管理情報ブロックに格納されている。このディレクトリポインタ1123の値は”2”である。また、このファイル管理情報ブロックの次エントリポインタ1121には”8”が設定されている。

【0188】また、ディレクトリCに対するディレクトリポインタ1123は、ファイル管理テーブル112”のエントリ”8”のファイル管理情報ブロックに格納されている。このディレクトリポインタ1123の値は”1”である。また、このファイル管理情報ブロックの次エントリポインタ1121には”9”が設定されている。

【0189】また、ファイルEに対するディレクトリポインタ1123は、ファイル管理テーブル112”のエントリ”9”のファイル管理情報ブロックに格納されている。このディレクトリポインタ1123の値は”8”である。また、このファイル管理情報ブロックの次エントリポインタ1121には”10”が設定されている。

【0190】また、ファイルFに対するディレクトリポインタ1123は、ファイル管理テーブル112”のエントリ”10”のファイル管理情報ブロックに格納されている。このディレクトリポインタ1123の値は”8”である。また、このファイル管理情報ブロックの次エントリポインタ1121には”11”が設定されている。

【0191】また、ファイルGに対するディレクトリポインタ1123は、ファイル管理テーブル112”のエ

ントリ”11”のファイル管理情報ブロックに格納されている。このディレクトリポインタ1123の値は”8”である。また、このファイル管理情報ブロックの次エントリポインタ1121には”13”が設定されている。

【0192】また、ファイルAに対するディレクトリポインタ1123は、ファイル管理テーブル112”のエントリ”13”のファイル管理情報ブロックに格納されている。このディレクトリポインタ1123の値は”1”である。また、このファイル管理情報ブロックの次エントリポインタ1121には”14”が設定されている。

【0193】また、ファイルBに対するディレクトリポインタ1123は、ファイル管理テーブル112”のエントリ”14”のファイル管理情報ブロックに格納されている。このディレクトリポインタ1123の値は”1”である。また、このファイル管理情報ブロックの次エントリポインタ1121には”0”が設定されている。

【0194】このように、ディレクトリポインタ1123には、木構造において自己のすぐ上に位置するディレクトリのファイル管理テーブル112”内のエントリの番号が設定される。また、次エントリポインタ1121には、ファイル管理テーブル112”中の使用されているエントリ同士、あるいは未使用なエントリ同士を連結するように、ディレクトリまたはファイルのファイル管理テーブル112”内のエントリの番号が設定される。

【0195】また、ファイルシステム管理ブロック113”においては、使用済みエントリ先頭ポインタ113b’にファイル管理テーブル112”のエントリ”1”が設定されている。また、使用済みエントリ最終ポインタ113c’にはファイル管理テーブル112”のエントリ”14”が設定されている。

【0196】さらに、未使用エントリ先頭ポインタ113dには、ファイル管理テーブル112”内のエントリ”5”が設定されている。また、未使用エントリ最終ポインタ113eには、ファイル管理テーブル112”内のエントリ”15”が設定されている。

【0197】このため、ファイル管理テーブル112”のエントリ”5”の次エントリポインタ1121には”6”が設定されている。また、ファイル管理テーブル112”のエントリ”6”の次エントリポインタ1121には”7”が設定されている。さらに、ファイル管理テーブル112”のエントリ”7”の次エントリポインタ1121には”12”が設定されている。また、さらに、ファイル管理テーブル112”のエントリ”12”の次エントリポインタ1121には”15”が設定されている。そして、ファイル管理テーブル112”のエントリ”15”の次エントリポインタ1121には”0”が設定されている。

【0198】また、ファイル管理テーブル112”の各エントリには、図19に示すファイルシステム110内のデータ領域内に作成される該当するディレクトリのデータ構造(ディレクトリファイル)へのポインタも格納されている。

【0199】図21は、ディレクトリエントリ削除部134によって実行される、あるファイルを削除する際の処理を説明するフローチャートである。まず、あるファイルを削除する際、そのファイルの存在するディレクトリファイルのファイル管理テーブル112”中でのエントリ位置をディレクトリポインタ1123によって知ることができる。次に、そのディレクトリファイルのデータブロックをオープンし、先頭から順に、そのディレクトリファイル内のエントリを順次、読み出し、上記削除対象のファイルのエントリを見つけ出す(S301～S304)。

【0200】そして、削除対象のファイルのエントリを見つけたら(S304, YES)、そのエントリを参照して、そのディレクトリファイルのデータブロックからそのファイルのデータを削除する(S305)。

【0201】図22は、第一の基本構成の第4の実施例のシステム構成図である。この第4の実施例のシステム構成は、ほぼ、図17のシステムと同様であり、ファイルシステム管理機構130eに削除候補ファイル問い合わせ部131eが追加されている。

【0202】図23は、この削除候補ファイル問い合わせ部131eの動作を説明するフローチャートである。削除候補ファイル問い合わせ部131eは、アプリケーションプログラムが削除候補ファイル問い合わせ用のシステムコールを実行すると、図24に示す削除候補ファイル問い合わせフラグを受け取る。そして、その削除候補ファイル問い合わせフラグの値が”1”であるか否かを判別する(S401)。

【0203】図24に示すように、削除候補ファイル問い合わせフラグの値が”1”であれば、これは、削除候補ファイルに関する情報の問い合わせを意味する。したがって、削除候補ファイル問い合わせフラグの値が”1”であれば(S401, YES)、削除候補ファイルがあるか否かをファイルシステム管理ブロック113”とファイル管理テーブル112”を参照して調べる(S402)。

【0204】そして、削除候補ファイルがあれば、そのファイルに関する情報(作成日時、アクセス日時等)をアプリケーションプログラムに通知する。このとき、例えば該アプリケーションプログラムは、通知された削除候補ファイルの一覧を所定の表示装置等に表示させる(S403)。

【0205】次に、図25は、図1の削除順序問い合わせ/設定部136の動作の一例を説明するフローチャートである。この削除順序問い合わせ/設定部136は、

アプリケーションプログラムから図26に示す削除順序問い合わせ/設定フラグを受け取る。

【0206】上記削除条件問い合わせ/設定部フラグは、値が“2”のとき「問い合わせ」、「3”のとき「設定」を意味する。すなわち、アプリケーションプログラムは、ファイルの削除順序を問い合わせる場合には上記削除順序問い合わせ/設定部フラグに“2”を設定し、ファイルの削除条件を変更したい場合には上記削除順序問い合わせ/設定部フラグに“3”を設定して、削除順序問い合わせ/設定部136に対してシステムコールを行う。

【0207】また、削除順序情報管理部133は、削除順序情報格納部114にファイル削除順序情報を記憶している。図27は、削除順序情報管理部133が削除順序情報格納部114に登録するファイルの削除順序情報の一覧を示す図である。

【0208】このファイルの削除順序情報は、8進数で示され、“01”、“02”、“04”、“010”、“020”及び“0200”の各値を有する。これらの各8進数値は、以下のファイル削除順序情報に対応している。

【0209】

“01”：「作成日時順」

“02”：「更新日時順」

“04”：「アクセス日時順」

“010”：「サイズ順」

“020”：「アクセス回数順」

“0200”：「逆順」

図25のフローチャートの説明をする。

【0210】削除順序問い合わせ/設定部136は、アプリケーションプログラムから削除順序問い合わせ設定のシステムコールを受け取ると、そのシステムコールのパラメータである削除順序問い合わせ設定フラグの値を調べ、アプリケーションプログラムからの要求が「問い合わせ」であるか否かを判別する(S501)。

【0211】そして、削除順序問い合わせ設定フラグの値が“2”であれば、「問い合わせ」であると判別し(YES)、削除順序情報管理部133に対して、現在、設定されている削除順序情報を問い合わせる。削除順序情報管理部133は、削除順序情報格納部114を参照して、現在、設定されている削除順序情報を削除順序問い合わせ/設定部136に返す。削除順序問い合わせ/設定部136は、この削除順序情報をアプリケーションプログラムに通知する。このとき、例えば、該アプリケーションプログラムは、この通知されたファイルの削除順序情報を所定の表示装置に表示させる(S505)。

【0212】一方、上記ステップS501で削除順序問い合わせ設定フラグの値が“2”でなければ(NO)、次に、削除順序問い合わせ設定フラグの値が“3”、す

なわち、「設定」の指定であるか否かを判別する(S502)。そして、削除順序問い合わせ設定フラグの値が“3”、すなわち、「設定」の指定であれば(YES)、削除順序情報管理部133に依頼して削除順序情報格納部114に登録されるファイルの削除順序情報の書換えを行う(S503)。

【0213】続いて、その削除順序フラグの値に従って、ファイル管理テーブル112内のファイルの削除順序のリストを変更する(S504)。次に、述べる実施例は、図1のファイル削除部132、削除条件判断部137、削除条件問い合わせ/設定部138及び削除条件情報管理部139によって行われる、ある条件設定を満足するファイルのみを削除するものである。

【0214】削除条件情報管理部139は、図28に示す削除条件フラグ(mode)に関する情報を、削除条件情報格納部116に登録・管理している。削除条件フラグ(mode)は、“1”～“8”の値が有効であり、各値は、以下のような削除条件に対応している。

【0215】

“1”：作成日時

“2”：更新日時

“3”：アクセス日時

“4”：サイズ

“5”：ユーザ名

“6”：グループ名

“7”：リンク数

“8”：タイプ

また、削除条件情報管理部139は、図29に示す削除条件フラグ(type)に関する情報を管理している。

【0216】削除条件フラグ(type)は、“1”～“5”の値が有効であり、各値は、以下のような削除条件に対応している。

“1”：通常ファイル

“2”：ディレクトリ

“3”：キャラクタスペシャルファイル

“4”：ブロックスペシャルファイル

“5”：シンボリックリンクファイル

また、さらに、削除条件情報管理部139は、図30に示す削除条件フラグ(cond)に関する情報を管理している。

【0217】削除条件フラグ(cond)は、“1”～“6”の値が有効であり、各値は、以下のような削除条件に対応している。

“1”：EQ(等しい)

“2”：GT(大きい)

“3”：LT(小さい)

“4”：GE(以上)

“5”：LE(以下)

“6”：NE(等しくない)

また、削除条件問い合わせ/設定部138は、図31に

示す削除条件問い合わせ/設定部フラグに関する情報を記憶している。

【0218】この削除条件問い合わせ/設定部フラグは、“4”と“5”の値が有効であり、これらの値は、以下のような意味を有する。

“4”：問い合わせ

“5”：設定

アプリケーションプログラムは、上記3種類の削除条件フラグの値を設定して、ファイル削除部132に対してシステムコールを行うことにより、特定の条件を満足するファイルのみをファイルシステム110から削除することができる。

【0219】例えば、ディレクトリを削除したくない場合には、削除条件フラグ(cond)に“6”(NE)、削除条件フラグ(mode)に“8”(タイプ)、削除条件フラグ(type)に“2”(ディレクトリ)を設定して、システムコールを行う。

【0220】図32～図33は、ファイル削除部132が削除条件情報格納部116に登録されているファイルの削除条件情報を考慮して、ファイルを削除する処理の概略を説明するフローチャートである。

【0221】ファイル削除部132は、削除対象のファイルを見つけると(S601)、削除条件情報管理部139に、現在、削除条件情報格納部116に登録されているファイルの削除条件情報を問い合わせる。削除条件情報管理部139は、削除条件情報格納部116を検索して、現在、ファイルの削除条件が設定されているかを調べる(S602)。

【0222】そして、ファイルの削除条件が設定されていなければ(S602、NO)、そのファイルを削除する(S603)。一方、ファイルの削除条件が設定されていれば、その削除条件フラグ(mode)が何であるかを調べる(S604～S611)。すなわち、図28に示す“1”～“8”のいずれの値であるかを調べる。

【0223】そして、削除条件フラグ(mode)が「作成日時」であれば(S604、YES)、上記削除対象のファイルの作成日時情報を取得し、これを変数xに代入する。また、指定された作成日時を変数yに代入する(S612)。

【0224】また、削除条件フラグ(mode)が「更新日時」であれば(S605、YES)、上記削除対象のファイルの更新日時情報を取得し、これを変数xに代入する。また、指定された更新日時を変数yに代入する(S613)。

【0225】また、さらに、削除条件フラグ(mode)が「アクセス日時」であれば(S606、YES)、上記削除対象のファイルのアクセス日時情報を取得し、これを変数xに代入する。また、指定されたアクセス日時を変数yに代入する(S614)。

【0226】次に、削除条件フラグ(cond)が何であるかを調べる(S620～S625)。そして、削除条件フラグcondが「EQ」であれば(S620、YES)、変数xと変数yが等しいかを調べ(S626)、等しければ(YES)、上記削除対象のファイルを削除する(S627)。

【0227】一方、変数xと変数yが等しくなければ(S626、NO)、次の削除対象のファイルを見つけ(S628)、上記ステップS602に戻る。図34は、本発明の第二の基本構成のシステム構成を示すブロック図である。

【0228】同図において、図1に示すブロックと同一のブロックには同一の符号を付与している。この第二の基本構成が、前述した図1の第一の基本構成と異なる所は、3つの削除順序情報格納部145-1～145-3を備えていることである。また、これらの各削除順序情報格納部145-1～145-3は、それぞれ、第一のファイル削除順序情報、第二のファイル削除順序情報、第三のファイル削除順序情報を格納している。

【0229】さらに、これらの各削除順序情報格納部145-1～145-3に対応して、削除順序情報管理部203-1～203-3が設けられている。これらの削除順序情報管理部203-1～203-3は、それぞれ、削除順序情報格納部145-1～145-3に格納されるファイル削除順序情報を登録・管理する。

【0230】さらに、上記削除条件情報管理部203-1～203-3を一元的に総合して管理する削除順序情報総合管理部201が設けられている。この削除順序情報総合管理部201は、削除条件問い合わせ/設定部136、削除候補ファイル問い合わせ部140、ファイル削除部132及び優先順位情報管理部202からの要求を受け付ける。

【0231】該優先順位情報管理部202は、優先順位情報格納部244に、上記3つの削除順序情報格納部145-1～145-3に格納されているファイルの削除順序情報の優先順位に関する情報を登録・格納する。

【0232】さらに、優先順位情報問い合わせ/設定部205が設けられている。この優先順位情報問い合わせ/設定部205は、優先順位情報管理部202に対して、上記3つの削除順序情報格納部145-1～145-3に格納されているファイル削除順序情報の優先順位情報を問い合わせたり、それらのファイル削除順序情報の優先順位情報の設定を依頼する。

【0233】この第二実施例の概略動作を説明する。ファイルの新規作成時に、ファイルシステム110に空き容量が不足したり、また、さらに、ファイルを削除することになった場合、ファイル削除部132は、削除順序情報総合管理部201に対して、削除対象のファイルの問い合わせを行う。削除順序情報総合管理部201は、優先順位情報管理部202を介して、優先順位情報格納

部244に登録されている上記3つの削除順序情報格納部145-1~145-3に格納されているファイル削除順序情報の優先順位情報を参照し、この参照結果に基づいて、3つの削除順序情報管理部203-1~203-3の中の1つまたは複数に対してファイル削除順序情報を問い合わせ、削除の対象となるファイルの情報を得る。

【0234】また、優先順位情報問い合わせ・設定部205は、優先順位情報管理部202に対して削除順序情報格納部145-1~145-3に格納されている複数のファイル削除順序情報の優先順位を問い合わせる。また、優先順位情報管理部202を介して、優先順位情報格納部244に3つの削除順序情報格納部145-1~145-3に格納されている3つのファイル削除順序情報の優先順位を設定する。

【0235】この第二実施例では、例えば、システムの資格で作成したファイルについては、作成日時の古い順から削除し、また、ユーザの資格で作成したファイルについては、サイズの大きい順に削除するようにすることができる。また、ファイルを削除する際、まず、ユーザの資格で作成されたファイルを先に削除し、次に、システムの資格で作成されたファイルを削除するようにすることも可能である。

【0236】図35は、ファイル削除部132がファイルを削除する必要が生じた場合に、優先順位情報管理部202に対してファイルの削除順序情報の優先順位を問い合わせたり、設定するために、また、その優先順位情報の数を問い合わせたり、設定するために使用するファイルの削除順序情報の優先順位問い合わせ/設定フラグを説明する図である。

【0237】このファイルの削除順序情報の優先順位問い合わせ/設定フラグの値は、以下のような意味を有している。

”6”：ファイルの削除順序情報の優先順位の問い合わせ

”7”：ファイルの削除順序情報の優先順位の設定

”8”：優先順位情報の個数の問い合わせ

”9”：優先順位情報の個数の設定

ファイル削除部132は、ファイル作成/削除機構131bから削除対象のファイルを見つけて欲しいという要求を受け取ると、削除順序情報総管理部201に対して削除対象のファイルを問い合わせる。

【0238】削除順序情報総管理部201は、この問い合わせを受けて、図36に示すフローチャートの処理を実行して、削除対象ファイルを決定する。削除順序情報総管理部201は、まず、優先順位情報管理部202を介してファイルの削除順序情報の優先順位情報を取得する。次に、その優先順位情報を基に、まず、最も優先順位の高いファイルの削除順序情報を、3つの削除順序情報管理部203-1~203-3のいずれかを介し

て取得する(S701)。

【0239】次に、未検索のファイルの削除順序情報があるか否かを判別する(S702)。この場合、最も優先順位の高いファイルの削除順序情報を取得したので、続いて、そのファイルの削除順序情報の中で最も削除順位の早い削除候補ファイルを選択する(S703)。

【0240】次に、上記ファイルの削除順序情報の中に削除候補ファイルがまだ残っているか否かを判別する(S704)。この場合、削除候補ファイルを選択したので、続いて、この削除候補ファイルが削除条件に一致しているか否かを判別する(S706)。そして、削除条件に一致していれば(S706, YES)、その削除候補ファイルを削除対象のファイルとして決定する(S708)。

【0241】一方、削除条件に一致していなければ、上記ファイルの削除順序情報の中から次に削除順序の早いファイルを選択する(S707)。そして、ステップ704に戻る。

【0242】このようにして、選択したファイルの削除順序情報の中から削除順に削除条件に一致する削除候補ファイルを検索していく。そして、選択したファイルの削除順序情報の中に削除条件に一致する削除候補ファイルが一つも無ければ、上記ステップS701と同様な処理により、次に優先順位の高いファイルの削除順序情報を取得する(S705)。

【0243】そして、このファイルの削除順序情報に対して、上記ステップS702~S708の処理を実行する。このようにして、削除条件に一致している削除候補ファイルが見つかるまで、ファイルの削除順位情報を優先順位順に順次選択し、そのファイルの削除順位情報の中から削除条件に一致している削除対象ファイルを見つけて出す。図37は、本発明の第7の実施例のシステム構成を示すブロック図である。同図において、ファイルデータブロック301は、ファイルの実データを格納しているブロックである。ファイル実サイズ情報部302は、ファイルデータブロック301の実サイズを記憶する。先頭位置保持部303は、ファイルデータブロック301の先頭位置を記憶する。最終位置保持部304は、ファイルデータブロック301の最終位置を記憶する。サイズ上限値保持部305は、ファイルデータブロック301のサイズの上限を記憶する。比較部306は、ファイル実サイズ情報部302に保持されているサイズとサイズ上限値保持部305に保持されているサイズとを比較し、その比較結果を重ね書き指示部308に出力する。

【0244】単位解析部307は、設定者がファイルデータブロック301のサイズを、例えば、バイト、ブロック、Kバイト(キロバイト)などの単位で指定する際、この単位を解析して、該設定者が指定したファイルデータブロック301の上限値をサイズ上限値保持部3

05に設定する。重ね書き指示部308は、比較部306によりファイルデータブロック301の上限までデータが書き込まれたことが検出されると、ファイルデータブロック301に対してデータを重ね書きするように指示する。

【0245】次に、この第七の実施例の第一の動作例を、図38及び図39を参照しながら説明する。データが全く書き込まれていない状態のときには、先頭位置保持部303と最終位置保持部304は、共に、ファイルデータブロック301の先頭位置を指している。また、ファイル実サイズ情報部302は"0"に設定されている(図39(a)の状態1)。

【0246】この後、ファイルデータブロック301にデータの書き込みが行われると、先頭位置保持部303はファイルデータブロック301の先頭を指したまま変化しないが、最終位置保持部304はファイルデータブロック301に書き込まれたデータ量に従って、ファイルデータブロック301に書き込まれたデータの最終位置を指しながら移動する。また、ファイル実サイズ情報部302は、ファイルデータブロック301に書き込まれたデータ量を常に示すように変化する(図39(b)の状態2)。

【0247】そして、ファイルデータブロック301が図38(c)に示す状態3になったとき、重ね書き指示部308が、不図示のデータ書き込み部に対して、ファイルデータブロック301に対するデータの書き込みを、重ね書きにより行うように指示すると、以後のファイルデータブロック301に対するデータの書き込みは、上書きによって行われる。すなわち、ファイルデータブロック301が上記状態3にあるとき、前記データ書き込み部がファイルデータブロック301に対してデータを追記しようとする、最終位置保持部304は、ファイルデータブロック301の先頭位置を指すように戻り、最初に書き込んだデータに重ね書きを行う(図39(d)の状態4参照)。

【0248】そして、以後、ファイルデータブロック301に対するデータの書き込みは重ね書きによって行われる。また、これ以後、ファイルデータブロック301に対してデータの書き込みが行われても、ファイル実サイズ情報部302の値は、変化せず、常に、一定の値(状態3のときのファイルデータブロック301のデータサイズ)を示している。

【0249】最終位置保持部304は、実際のデータの最終書き込み位置を指すように移動する。また、先頭位置保持部303も、最終位置保持部304と同一の位置を指すように移動する(図39(e)の状態5)。さらに、ファイルデータブロック301に対してデータの書き込みが行われ、再び、データがファイルデータブロック301の最終位置まで書き込まれると(図39(f)の状態6)、先頭位置保持部303と最終位置保持部304は、共に、ファイルデータブロック301の先頭の位置を指すようになる(図38(d)の状態4)。

【0250】そして、さらに、ファイルデータブロック301に対して、データが書き込まれると、先頭位置保持部303と最終位置保持部304は、共に、ファイルデータブロック301の先頭位置を指すようになった後、以下、同様の動作を繰り返す。

【0251】次に、この第七の実施例の第二の動作例を、図40を参照しながら説明する。この例では、予め、サイズ上限値保持部305にファイルデータブロック301のサイズの上限值を設定しておく(図40(a)の状態1)。

【0252】ファイルデータブロック301にデータが書き込まれる際、比較部306はファイル実サイズ情報部302の値とサイズ上限値保持部305の値を比較する。そして、両者の値が等しくなったことを検出すると(図40(b)の状態2)、重ね書き指示部308にこのことを通知する。すなわち、ファイルデータブロック301のサイズの上限までデータが書き込まれると、比較部306は、これを重ね書き指示部308に通知する。

【0253】重ね書き指示部308は、この通知を受け取ると、データ書き込み部に重ね書きを指示する。これ以降の動作は、上記第一の動作例において、重ね書き指示部308からデータの重ね書きの指示がなされたときと同様である。

【0254】次に、第七の実施例の第三の動作例を説明する。ファイルデータブロック301の作成者は、ファイルデータブロック301の上限値のサイズをバイト、ブロック、Kバイト(キロバイト)などの所定の単位で指定することができる。単位解析部307は、この指定される単位の意味を解析して、サイズ上限値保持部305に正しいファイルデータブロック301のサイズの上限值を設定する。

【0255】次に、図41は、データ書き込み部によって実行される上記第七の実施例の第一の動作例の処理を説明するフローチャートである。尚、ファイル作成時には、先頭位置保持部303と最終位置保持部304の値は、共に、"0"(ファイルデータブロック301の先頭位置)に設定される。また、ファイル実サイズ情報部302には、"0"(サイズが"0")に設定される(図38(a)の状態1)。

【0256】まず、ファイルデータブロック301に書き込むべきデータがあるか否かを判別し(S711)、書き込むべきデータがあれば(YES)、次に、重ね書き指示部308からデータの重ね書きの指示がなされているか否かを判別する(S712)。

【0257】そして、データの重ね書きの指示がなされていないならば(NO)、ファイルデータブロック301に1バイトのデータを書き込む(S713)。続いて、

ファイル実サイズ情報部302の値を”1”増加させ（S714）、次に、最終位置保持部304の値を”1”増加させる（S715）。

【0258】以上の動作が繰り返されて、ファイルデータブロック301には、データが1バイトずつ書き込まれていく。そして、ファイルデータブロック301が図38（c）の状態3にあるときに、重ね書き指示部308からデータの重ね書きの指示がなされると、ステップS712で、データの重ね書きの指示がなされていると判別し（YES）、次に、最終位置保持部304の値がファイル実サイズ情報部302の値以上であるか否かが判別される（S716）。状態3のとき、両者の値は等しいので（YES）、最終位置保持部304に”0”を設定する（S717）。この結果、図38（d）に示す状態4になる。

【0259】続いて、先頭位置情報保持部303と最終位置保持部304の値が等しいか否かを判別する（S718）。状態4のときには、両者の値が等しいので、（YES）、先頭位置情報保持部303の値に”1”を加える（S719）。

【0260】次に、先頭位置保持部303の値がファイル実サイズ情報部302の値以上であるか否かを判別する（S720）。この判別において、図38（d）の状態4のときには、先頭位置情報保持部303の値がファイル実サイズ情報部302の値以上ではない（NO）と判別される。

【0261】次に、ファイルデータブロック301の最終位置保持部304の指す位置に、1バイトのデータを書き込み（S722）、最後に最終位置保持部304の値に”1”を加える（S723）。このとき、先頭位置情報保持部303と最終位置保持部304の値は共に”1”となっている。

【0262】以後、処理はステップS711に戻る。そして、データの重ね書き指示がなされる毎に、ステップS711→S712→S718→S719→S720→S723の一連の処理が繰り返される。そして、この一連の処理が1回行われる毎に、先頭位置情報保持部303と最終位置保持部304の値は、”2”、”3”、”4”・・・と、順次”1”ずつ大きくなっていく（図39（e）の状態5参照）。以上の一連の処理は、先頭位置情報保持部303と最終位置保持部304の値が、ファイル実サイズ情報部302の値より小さい間、繰り返される。

【0263】そして、図39（f）の状態6の状態になったとき、ステップS720で先頭位置情報保持部303の値がファイル実サイズ情報部302の値以上であると判別し（YES）、先頭位置情報保持部303に”0”を設定する（S721）。

【0264】次に、ファイルデータブロック301の最終位置保持部304の指す位置に、1バイトのデータを

書き込み（S722）、続いて、最終位置保持部304の値に”1”を加える（S723）。このとき、先頭位置保持部303の値は”0”に、最終位置保持部304の値はファイル実サイズ情報部302と同じ値となっている。そして、次に、ファイルデータブロック301に1バイトのデータが書き込まれるときには、ステップS716で最終位置保持部304の値がファイル実サイズ情報部302の値以上であると判別され（YES）、この結果、最終位置保持部304には”0”が設定される（S717）。

【0265】以降は、再び、先頭位置情報保持部303と最終位置保持部304の値が共に”1”、”2”、”3”・・・と同じ値を取りながら、同様の処理が繰り返される。

【0266】尚、以上の説明では、先頭位置情報保持部303と最終位置保持部304の値が常に等しくなるため、ステップS718では常にYESと判別されるが、先頭位置情報保持部303の値が一度に”1”より多く増加した場合には、ステップS718でNOと判別される。このような場合の処理については、後述する。

【0267】次に、図42は、データ書き込み部によって実行される上記第七の実施例の第二の動作の処理を説明するフローチャートである。尚、ファイル作成時には、先頭位置保持部303と最終位置保持部304の値は、共に、”0”（ファイルデータブロック301の先頭位置）に、また、ファイル実サイズ情報部302は”0”（サイズが”0”）に設定される。また、サイズ上限値保持部305にファイルデータブロック301のサイズの上限の値が設定される。また、データの重ね書きを行っていることを示す重ね書き動作中フラグは”偽”（OFF）に設定される（図40の状態1参照）。

【0268】まず、データ書き込み部は、ファイルデータブロック301に書き込むデータがあるか否かを判別し（S801）、書き込むデータがあれば（S801、YES）、次に、重ね書き動作中フラグが真（ON）であるか否かを判別する（S802）。

【0269】初期状態では、データの重ね書き動作中フラグ”偽”（OFF）であるので、ステップS802でNOと判別され、ファイルデータブロック301の最終位置保持部304の指す位置に1バイトのデータを書き込む（S803）。続いて、ファイル実サイズ情報部302の値を”1”増加させ（S804）、次に、最終位置保持部304の値を”1”増加させる（S805）。

【0270】次に、ファイル実サイズ情報部302の値がサイズ上限値保持部305の値以上であるか否かを判別する（S806）。そして、ファイル実サイズ情報部302の値がサイズ上限値保持部305の値以上でない（S806、NO）、ステップS801に戻る。

【0271】ステップS806で、ファイル実サイズ情

報部302の値がサイズ上限値保持部305の値以上でない(NO)と判別されている間、上述したステップS801→S802→S803→S804→S805→S806の一連の処理が繰り返して実行され、ファイルデータブロック301にはデータが1バイトずつ書き込まれていく。また、ファイル実サイズ情報部302及び最終位置保持部304の値が、共に"1"ずつ増加していく。

【0272】そして、ファイルデータブロック301が図40の状態2の状態になると、ステップS806でYESと判別され、重ね書き動作中フラグが"真"に設定される。

【0273】この後、ステップS801の処理に戻り、ファイルデータブロック301に書き込むべきデータがある場合には(S801, YES)、ステップS802で重ね書き動作中フラグが"真"であると判別され、ステップS808の処理に移行する。

【0274】ステップS808～S815の処理は、上述したステップS716～S723の処理と同様である。次に、図43は、単位解析部307によって実行される上記第七の実施例の第三の動作の処理を説明するフローチャートである。

【0275】まず、ファイルデータブロック301のサイズの上限值が入力されると(S901)、そのサイズ情報をサイズと単位に分離し、該サイズを変数Xに単位を変数Uに代入する(S902)。

【0276】すなわち、例えば、入力されたサイズ情報が100Kバイトであった場合、サイズは'100'、単位は'K'となる。また、この場合、サイズはバイトを基本とし、これ以外に、Kバイト、Mバイトでの指定が可能であり、また、ブロック単位での指定も可能であるとする(1ブロックは512バイトとする)。バイトで指定された場合、単位は"無し"となる。

【0277】次に、変数Uの内容をステップS903、S905、S907、S909で解析する。そして、ステップS903で変数Uの内容が"無し"であると判別すれば、変数Yに"1"を代入する(S904)。また、ステップS905で変数Uの内容が"K"であると判別すれば、変数Yに"1024"を代入する(S906)。また、ステップS907で変数Uの内容が"B"であると判別すれば、変数Yに"512"を代入する(S908)。さらに、ステップS909で変数Uの内容が"M"であると判別すれば、変数Yに"1024×1024"を代入する(S910)。また、ステップS909で変数Uの内容が"M"でないとは判別すれば、バイト指定であると判断して、変数Yに"1"を代入する。

【0278】上記ステップS903～S911の処理が終了した後、変数Xと変数Yの値の乗算結果を変数Xに代入する(S912)。そして、変数Xの値をサイズ上

限值保持部305に設定する(S913)。

【0279】以上の結果、サイズ上限値保持部305にファイルデータブロック301の上限サイズがバイト単位で設定される。次に、図44は、本発明の第8の実施例のシステム構成図である。

【0280】ファイルシステム401は、ファイルデータブロック402、先頭位置保持部403、最終位置保持部404、空き容量下限保持部405、空き容量情報部406、空き容量判定部407、重ね書き指示部408、重ね書きフラグ409及びファイル実サイズ情報部410を備えている。

【0281】空き容量下限保持部405は、ファイルデータブロック402の空き容量の下限値が設定される。空き容量情報部406は、ファイルデータブロック402の現在の空き容量を記憶する。空き容量判定部407は、空き容量下限保持部405の値と空き容量情報部406の値とを比較し、ファイルデータブロック402の空き容量が空き容量下限保持部405に設定されているファイルデータブロック402の空き容量の下限値に等しくなったことを検出する。そして、検出すると、重ね書き指示部408にファイルデータブロック402に対するデータの重ね書きを指示する。先頭位置保持部403、最終位置保持部404及びファイル実サイズ情報部410は、上記第七実施例の先頭位置保持部303、最終位置保持部304及びファイル実サイズ情報部302と同様の機能を有する。

【0282】次に、この第8の実施例の第一の動作の概略を説明する。まず、予め、空き容量下限保持部405にファイルデータブロック402の空き容量の下限値を設定しておく。ファイルデータブロック402にデータを書き込んでいくと、次第に、ファイルデータブロック402の空き容量が減少していく。この空き容量は、空き容量情報部406に設定される。そして、ファイルデータブロック402の空き容量が空き容量下限保持部405に設定されている値以下になると、このことが、空き容量判定部407によって検出され、空き容量判定部407から重ね書き指示部408にこの検出が通知される。重ね書き指示部408は、この通知を受けて、不図示のデータ書き込み部にデータの重ね書きを指示する。上記以外の動作は、前記第七の実施例の第一の動作と同様である。

【0283】次に、この第8の実施例の第二の動作の概略を説明する。この第二の動作が、上記第一の動作と異なる点は、重ね書きフラグ409が「重ね書き可能」(ON)に設定されている場合のみ、ファイルデータブロック402に対してデータの重ね書きを行い、重ね書きフラグ409が「重ね書き不可」(OFF)に設定されている場合には、ファイルデータブロック402に対してデータの重ね書きを行わないことである。

【0284】図45は、この第二の動作におけるデータ

書き込み部の動作を説明するフローチャートである。尚、ファイル作成時には、先頭位置情報保持部403と最終位置保持部404は、“0”（ファイルデータブロック402の先頭位置）を指す値に設定される。また、ファイル実サイズ情報部410は、“0”（サイズが“0”）に設定される。さらに、空き容量下限保持部405にファイルデータブロック402の空き容量の下限値が設定される。そして、データの重ね書きを行っていることを示す重ね書き動作中フラグは“偽”（OFF）に設定される。

【0285】まず、データ書き込み部は、ファイルデータブロック402に書き込むべきデータがあるか否かを判別し（S921）、書き込むべきデータがあれば（YES）、次に、重ね書き動作中フラグが“真”（ON）か否かを判別する（S922）。

【0286】初期状態では、重ね書き動作中フラグは“偽”である（NO）ので、ファイルデータブロック402に1バイトのデータを書き込む（S923）。続いて、ファイル実サイズ情報部410の値を“1”増加させ（S924）、次に、最終位置保持部404の値を“1”増加させる（S925）。次に、重ね書き可能フラグ409が“ON”となっているか否かを判別し（S926）、その判別結果が“NO”であれば、ステップS921の処理に戻る。このように、重ね書き可能フラグ409が“OFF”に設定されていれば、ファイルデータブロック402に対してデータの重ね書きは行われぬ。

【0287】一方、上記ステップS926で重ね書き可能フラグ409が“ON”に設定されている（YES）と判別すると、次に、空き容量情報部406に設定されているファイルシステム401の空き容量の値が空き容量下限保持部405に保持されている値以上であるか否かを判別する（S927）。そして、ファイルシステムの空き容量の値が空き容量下限保持部405に設定されている値以下でない（NO）と判別すると、ステップS921の処理に戻る。

【0288】このようにして、ステップS927でファイルシステム401の空き容量の値が空き容量下限保持部405に設定されている値以下でないと判別されている間、ステップS921→S922→S923→S924→S925→S926→S927の一連の処理が繰り返行われ、ファイルデータブロック402にはデータが1バイトづつ書き込まれて行き、また、ファイル実サイズ情報部410と最終位置保持部404の値が共に“1”づつ増加していく。

【0289】そして、やがて、ステップS927でファイルシステム401の空き容量の値が空き容量下限保持部405の値以下である（YES）と判別されると、重ね書き動作中フラグが“真”に設定される（S928）。

【0290】その後、ステップS921の処理に戻り、次に、ステップS922で重ね書き動作中フラグ“真”である（YES）と判別され、この後、ステップS929以降の処理が行われる。このステップS929以降の処理は、上述した図42のフローチャートのステップS716～S723の処理と同様である。

【0291】次に、本発明の第九実施例を説明する。図46は、この第九実施例のシステム構成図である。同図において、図37に示すブロックと同一のブロックには同じ符号を付与している。

【0292】データ構造情報部509は、ファイルデータブロック301に書き込まれる各データについてその構造に関する情報を記憶している。データ境界検出部510は、ファイルデータブロック301に対してデータの重ね書きが開始された際、データ構造情報部509に格納されている情報を参照して、重ね書きされたデータの最終位置より後ろに位置する上記データによって重ね書きされなかったデータの先頭位置を見つけ、この先頭位置を先頭位置保持部303に設定する。上記以外の動作は、上記第七実施例の動作と同様である。

【0293】この第九実施例の概略動作を、図46を参照しながら説明する。図46に示すファイルデータブロック301の状態1においては、サイズが異なる同一のデータ構造を有する5個のデータ1～データ5が書き込まれている。これらのデータのデータ構造はデータ構造情報部509に格納されている。

【0294】この状態1の後で、ファイルデータブロック301に対してデータ6が書き込まれたとする。このとき、まだ、ファイルデータブロック301のサイズの上限までデータが書き込まれていない。続いて、データ7がファイルデータブロック301に対して書き込まれるとき、データの書き込み途中で比較部306によって、ファイルデータブロック301のサイズの上限までデータの書き込みが行われたことが検出され、重ね書き指示部308によってデータ書き込み部にデータ7を重ね書きにより書き込むように指示がなされる。そして、図46の状態2に示すように、データ7がデータ1の一部に重ね書きされる。

【0295】このとき、最終位置保持部304はデータ7の最終位置を指すように設定されるが、先頭位置保持部303は、まだ、重ね書きされていないデータ2の先頭を指すように設定される。これによって、重ね書きされたことによって、意味の無くなったデータ1のデータ構造の後半部分がデータブロックを専有するという無駄を無くすることができる。

【0296】図47は、第九実施例におけるデータ書き込み部の動作を説明するフローチャートである。データが全く書き込まれていない状態のときには、先頭位置保持部303と最終位置保持部304は、共に、ファイルデータブロック301の先頭位置を指している。また、

ファイル実サイズ情報部302は"0" (サイズが"0") に設定される。また、サイズ上限値保持部305にファイルデータブロック301のサイズの上限值が設定される。さらに、ファイルデータブロック301に対してデータの重ね書きを行っていることを示す重ね書き動作中フラグは"偽" に設定される。

【0297】図47のフローチャートに示す処理は、図42のフローチャートのステップS810～S813の処理を、ステップS1009～S1010の処理に置き換えた以外は、図42のフローチャートに示す処理と同様である。

【0298】すなわち、先頭位置保持部303と最終位置保持部304の値が等しいか否かを判別し (S1009)、等しいと判別すると (YES)、先頭位置保持部303にファイルデータブロック402の新しい先頭位置を設定する (S1010)。この新しい先頭位置は、データ境界検出部510によって検出される。

【0299】次に、図48は、図47のステップS1010で、データ境界検出部510が先頭位置保持部303に設定する新たな先頭位置を、見つけ出す処理を説明するフローチャートである。

【0300】尚、図46に示すように、ファイルデータブロック301に書き込まれる各データの先頭には、必ず、データ境界記録3011が書き込まれるものとする。

【0301】データ境界検出部510は、まず、先頭位置保持部303の値に"1"を加えた値を変数Xに代入する (S1021)。次に、変数Xの値がファイル実サイズ情報部302の値以上であるか否かを判別する (S1022)。そして、変数Xの値がファイル実サイズ情報部302の値以上でない (NO) と判別した場合にはステップS1024の処理に進む。

【0302】次に、変数Xの指すデータブロックの内容がデータ境界記録に等しいか否かを判別し (S1024)、変数Xの指すデータブロックの内容がデータ境界記録に等しくない (NO) と判別した場合には、続いて、変数Xの値が先頭位置保持部303の値に等しいか否かを判別する。そして、変数Xの値が先頭位置保持部303の値に等しくない (NO) と判別すると、次に、変数Xの値を"1"増加させる (S1028)。

【0303】この後、ステップS1022の処理に戻り、ステップS1022→(S1023→) S1024→S1026→S1028の処理が繰り返される。尚、この一連の処理の途中で、ステップS1024において変数Xの値がファイル実サイズ情報部302の値以上である (YES) と判別した場合には、変数Xに"0"を設定する (S1023)。

【0304】また、さらに、上記一連の処理の途中で、ステップS1024において変数Xの指すデータブロックの内容がデータ境界記録の値に等しい (YES)

と判別した場合には、新先頭位置保持部に変数Xの値を設定し、処理を終了する。

【0305】さらに、また、ステップS1022→(S1023→) S1024→S1026→S1028の処理が繰り返され、ステップS1026で変数Xの値が先頭位置保持部303の値と等しい (YES) と判別された場合には、新先頭位置保持部に変数Xの値に"1"を加えた値を設定し (S1025)、処理を終了する。これは、データ境界記録が見つからなかった場合の処理である。

【0306】このようにして、新先頭位置保持部に図47のフローチャートのステップS1010で先頭位置保持部303に設定される新先頭位置が格納される。次に、図49は、ファイルデータブロック301からデータを読み出す場合の処理を説明するフローチャートである。

【0307】まず、先頭位置保持部303の値を変数Xに代入する (S1101)。次に、変数Xの示すデータブロックの内容を1バイト読み出す (S1102)。続いて、変数Xの値を"1"増加し (S1103)、次に、変数Xの値が最終位置保持部404の値に等しいか否かを判別する (S1104)。そして、変数Xの値が最終位置保持部304の値に等しくない (NO) と判別した場合には、次に、変数Xの値がファイル実サイズ情報部302の値以上であるか否かを判別する (S1105)。そして、変数Xの値がファイル実サイズ情報部302の値以上でない (NO) と判別した場合には、ステップS1002の処理に戻る。このようにして、ステップS1104で変数Xの値が最終位置保持部404の値に等しいと判別されるまでの間、ステップS1102→S1103→S1105→(S1106)の処理が繰り返される。この一連の処理の途中で、ステップS1105で、変数Xの値がファイル実サイズ情報部302の値以上である (YES) と判別された場合には、変数Xに"0"を設定する (S1106)。

【0308】そして、変数Xの値が最終位置保持部304の値に等しいと判別した場合には、処理を終了する。以上の動作により、ファイルデータブロック301に格納されている全てのデータが、読み出される。

【0309】

【発明の効果】以上説明したように、本発明によれば、ファイルシステムを格納する記憶装置に空き容量が無くなった場合に、高速に、随時、必要なだけの空き容量を確保することが可能になる。

【0310】また、常に、一定量のログデータを保存することが可能になる。また、ファイルシステムの空き容量を圧迫することなく、ログデータを保存することが可能になる。

【0311】さらに、ファイルシステムの空き容量がある程度以下まで減った場合には、自動的にファイルのデ

ータの重ね書き動作に入るようにして、ファイルシステムの空き容量を圧迫しないようなファイルシステムの運用が可能になる。

【図面の簡単な説明】

【図1】本発明の第一の基本構成を示すブロック図である。

【図2】図1の基本構成の第一の実施例を示すブロック図である。

【図3】図2のシステムにおけるファイル作成時のファイル順序管理部の実行するファイル作成処理の動作を説明するフローチャートである。

【図4】図2のシステムにおけるファイル削除時のファイル順序管理部の実行するファイルの削除処理の動作を説明するフローチャートである。

【図5】図3、図4のフローにおける先頭及び最終ポイントの変化の様子を模式的に示した図である。

【図6】本発明の第一の基本構成の第二の実施例を示すブロック図である。

【図7】図6のファイル順序管理部によるファイル作成処理を説明するフローチャート（その1）である。

【図8】図6のファイル順序管理部によるファイル作成処理を説明するフローチャート（その2）である。

【図9】図6のファイル順序管理部による第1のファイル削除処理を説明するフローチャートである。

【図10】図6のファイル順序管理部による第2のファイル削除処理を説明するフローチャート（その1）である。

【図11】図6のファイル順序管理部による第2のファイル削除処理を説明するフローチャート（その2）である。

【図12】図6のファイル順序管理部によるファイル更新処理を説明するフローチャート（その1）である。

【図13】図6のファイル順序管理部によるファイル更新処理を説明するフローチャート（その2）である。

【図14】図7～図13における各ポイントの状態を模式的に示した図（その1）である。

【図15】図7～図13における各ポイントの状態を模式的に示した図（その2）である。

【図16】図7～図13における各ポイントの状態を模式的に示した図（その3）である。

【図17】本発明の第一の基本構成の第三の実施例を示すブロック図である。

【図18】ファイルシステムにおけるディレクトリの構成例を示す図である。

【図19】ファイルシステムのデータブロック内に作成される図18のディレクトリ構成に関する情報のデータ構造を示す図である。

【図20】図19に示すディレクトリのデータ構造を管理しているファイルシステム内のファイル管理テーブルとファイルシステム管理ブロックの構成図である。

【図21】ディレクトリエントリ削除部によって実行される、あるファイルを削除する際の処理を説明するフローチャートである。

【図22】本発明の第一の基本構成の第4の実施例のシステム構成図である。

【図23】削除候補ファイル問い合わせ部の動作を説明するフローチャートである。

【図24】削除候補ファイル情報問い合わせフラグの内容の一例を示す図である。

【図25】削除順序問い合わせ／設定部の動作の一例を説明するフローチャートである。

【図26】削除順序問い合わせ／設定フラグの内容の一例を示す図である。

【図27】削除順序情報管理部が削除条件情報格納部に登録するファイルの削除順序情報の一覧を示す図である。

【図28】削除条件フラグ（mode）の一覧を示す図である。

【図29】削除条件フラグ（type）の一覧を示す図である。

【図30】削除条件フラグ（cond）の一覧を示す図である。

【図31】削除条件問い合わせ／設定フラグの一例を示す図である。

【図32】ファイル削除部が削除条件情報格納部に登録されているファイルの削除条件情報を考慮して、ファイルを削除する処理の概略を説明するフローチャート（その1）である。

【図33】ファイル削除部が削除条件情報格納部に登録されているファイルの削除条件情報を考慮して、ファイルを削除する処理の概略を説明するフローチャート（その2）である。

【図34】本発明の第二の基本構成のシステム構成を示すブロック図である。

【図35】ファイルの削除順序情報の優先順位問い合わせ／設定フラグを説明する図である。

【図36】優先順位を考慮した削除候補ファイル検索処理を示すフローチャートである。

【図37】本発明の第七の実施例のシステム構成を示すブロック図である。

【図38】上記第七の実施例の第一の動作を説明する図（その1）である。

【図39】上記第七の実施例の第一の動作を説明する図（その2）である。

【図40】上記第七の実施例の第二の動作を説明する図である。

【図41】データ書き込み部によって実行される第七の実施例の第一の動作の処理を説明するフローチャートである。

【図42】データ書き込み部によって実行される第七の

実施例の第二の動作の処理を説明するフローチャートである。

【図43】単位解析部によって実行される第七の実施例の第三の動作の処理を説明するフローチャートである。

【図44】本発明の第8の実施例のシステム構成図である。

【図45】データ書き込み部の動作を説明するフローチャートである。

【図46】本発明の第九実施例のシステム構成図である。

【図47】本発明の第八の実施例におけるデータ書き込み部の動作を説明するフローチャートである。

【図48】データ境界検出部によって実行される図44のステップにおいて、先頭位置保持部に設定する新たな先頭位置を見つけだす処理を説明するフローチャートである。

【図49】ファイルデータブロックからデータを読み出す場合の処理を、説明するフローチャートである。

【符号の説明】

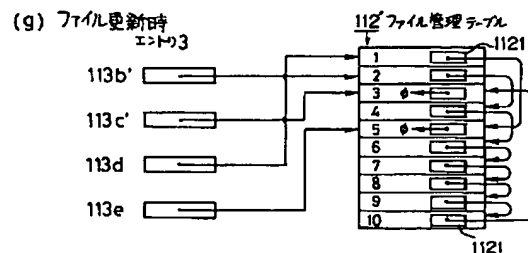
110 ファイルシステム
112 ファイル管理テーブル
130 ファイルシステム管理機構
131a 残量把握機構
131b ファイル作成／削除機構
132 ファイル削除部
133 削除順序情報管理部
134 ディレクトリエントリ削除部
135 ファイル／ディレクトリ対応情報管理部
136 削除順序問い合わせ／設定部
137 削除条件判断部
138 削除条件問い合わせ／設定部
139 削除条件情報管理部
140 削除候補ファイル問い合わせ部
141 削除順序情報

142 ファイル／ディレクトリ対応情報
143 削除条件情報
145-1 削除順序情報1
145-2 削除順序情報2
145-3 削除順序情報3
201 削除順序情報総合管理部
202 優先順位情報管理部
203-1 削除順序情報管理部1
203-2 削除順序情報管理部2
203-3 削除順序情報管理部3
205 優先順位情報問い合わせ／設定部
244 優先順位情報
301 ファイルデータブロック
302 ファイル実サイズ情報部
303 先頭位置保持部
304 最終位置保持部
305 サイズ上限保持部
306 比較部
307 単位解析部
308 重ね書き指示部
401 ファイルシステム
402 ファイルデータブロック
403 先頭位置保持部
404 最終位置保持部
405 空き容量下限保持部
406 空き容量情報部
407 空き容量判定部
408 重ね書き指示部
409 重ね書き可能フラグ
410 ファイル実サイズ情報部
509 データ構造情報部
510 データ境界検出部
3011 データ境界レコード

【図16】

【図24】

図7～図13における各ポインタの状態を模式的に示した図
(その3)

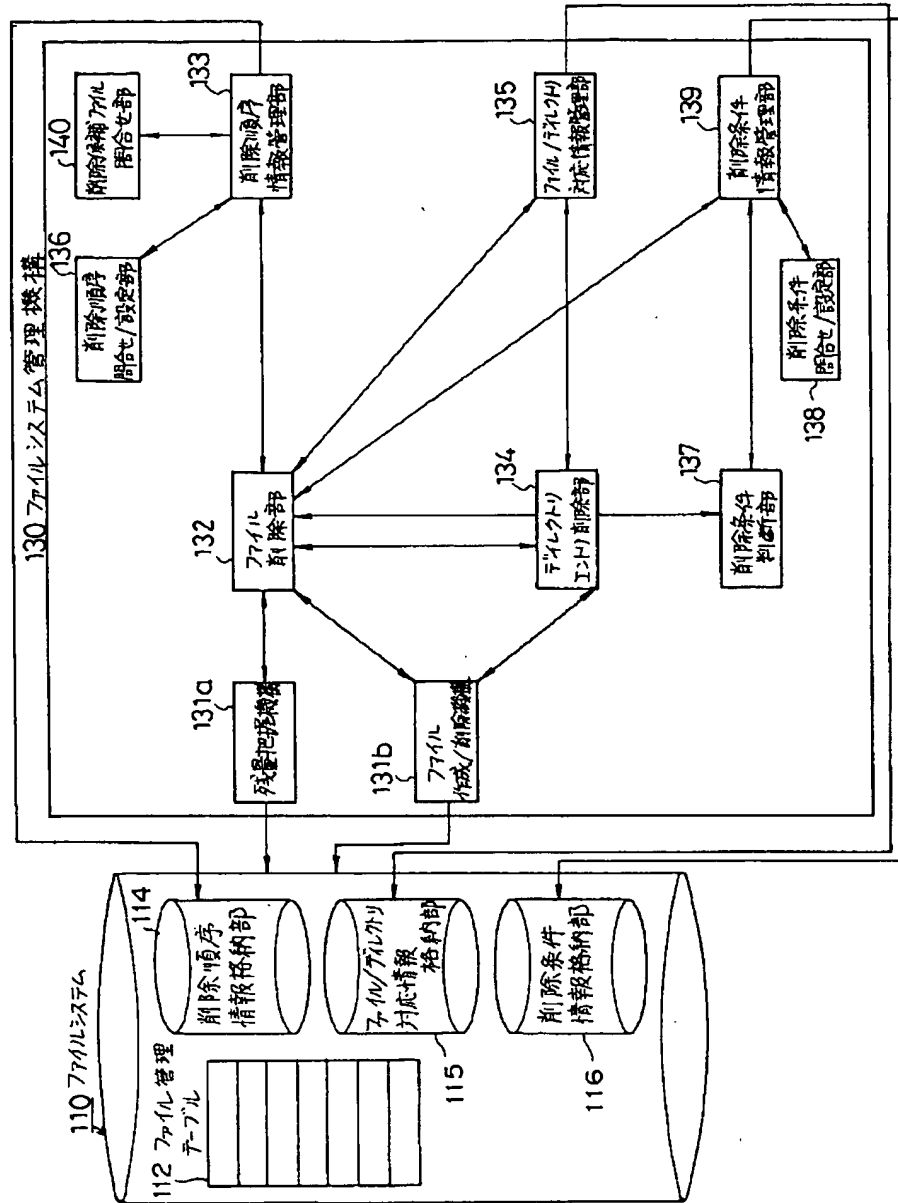


削除候補ファイル情報問い合わせ
フラグの内容の一例を示す図

| 意 味 | 値 |
|-------|---|
| 問 合 せ | 1 |

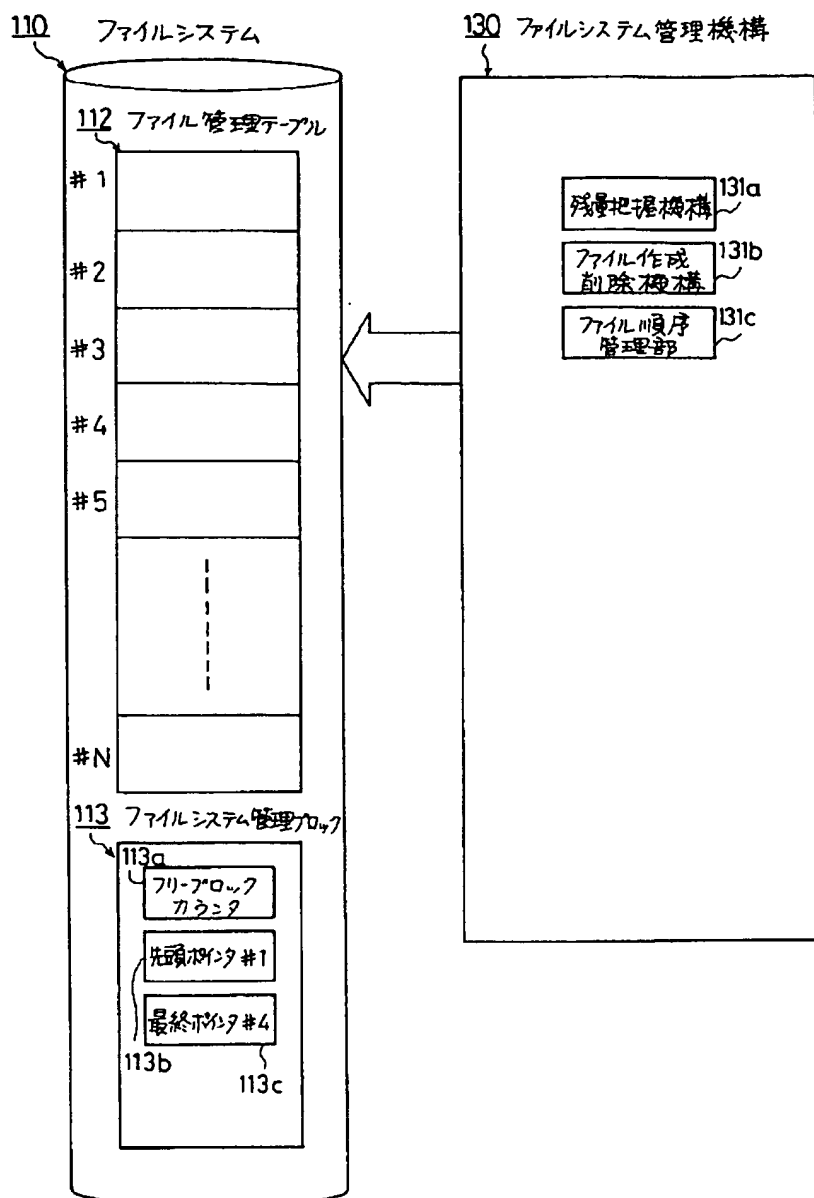
【図1】

本発明の第一の基本構成を示すブロック図



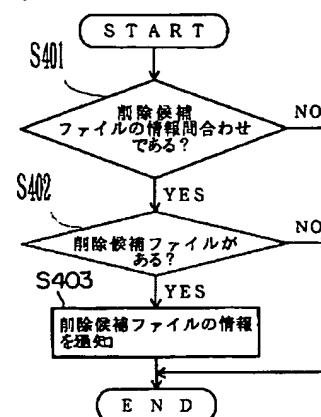
【図2】

図1の基本構成の第一の実施例を示すブロック図



【図23】

削除候補ファイル問い合わせ部の動作を説明するフローチャート



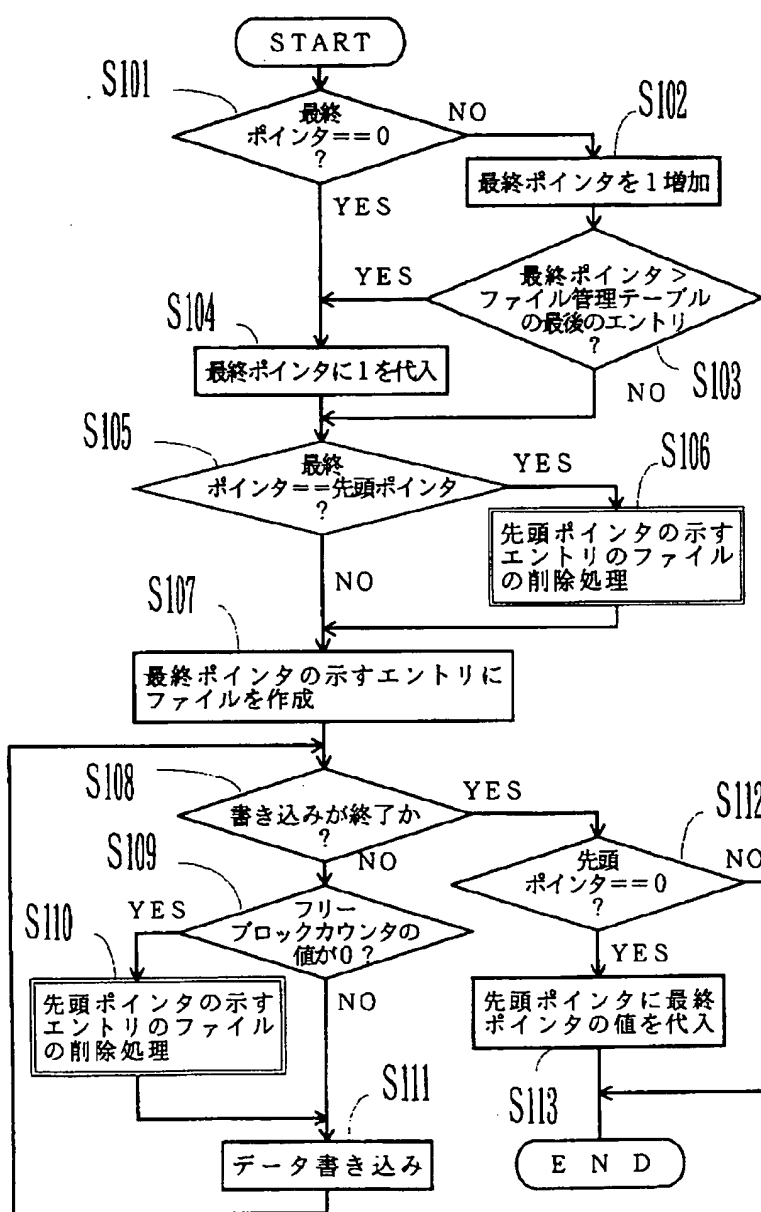
【図31】

削除条件問合せ/設定フラグの一例を示す図

| 意味 | 値 |
|-----|---|
| 問合せ | 4 |
| 設定 | 5 |

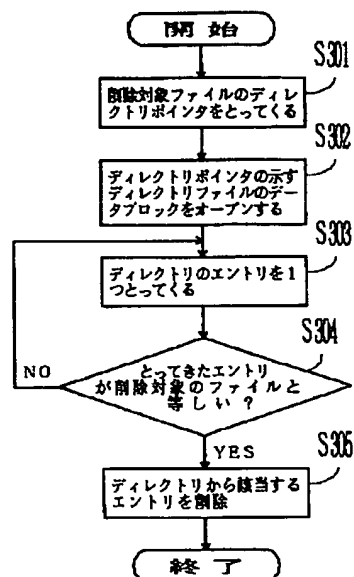
【図3】

図2のシステムにおけるファイル作成時のファイル順序管理部の
実行するファイル作成処理の動作を説明するフローチャート



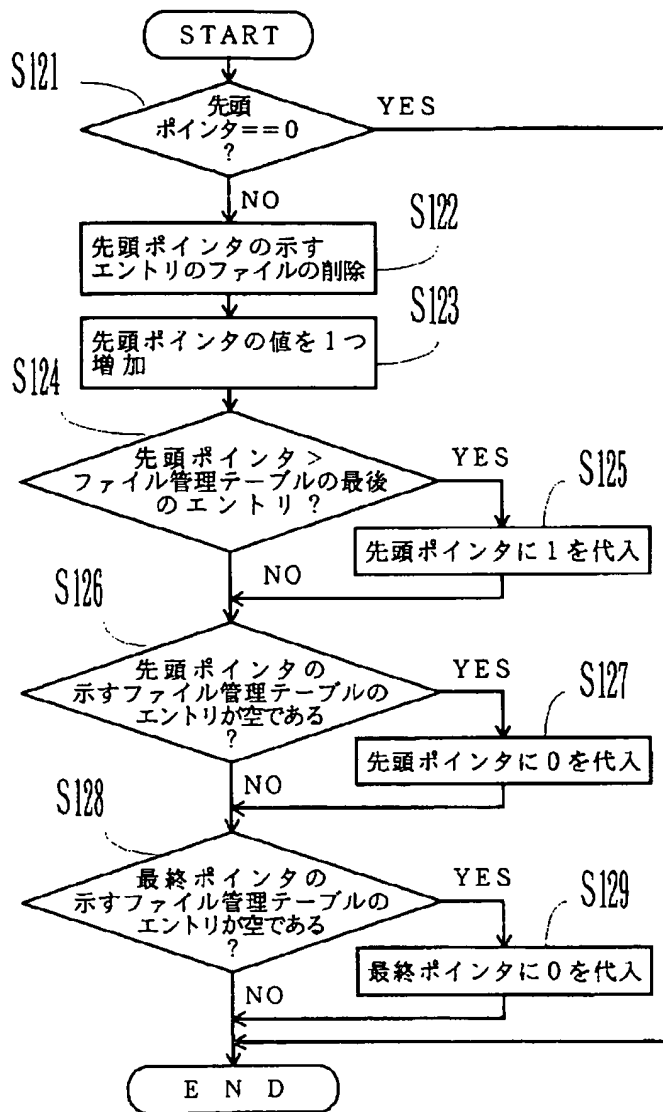
【図21】

ディレクトリエン트리削除部によって
実行されるあるファイルを削除する際の
処理を説明するフローチャート



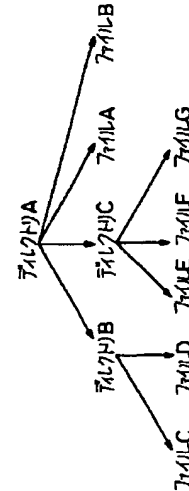
【図4】

図2のシステムにおけるファイル削除時のファイル順序管理部
の実行するファイル削除処理の動作を説明するフローチャート



【図18】

ファイルシステムにおけるディレクトリの
構成例を示す図



【図26】

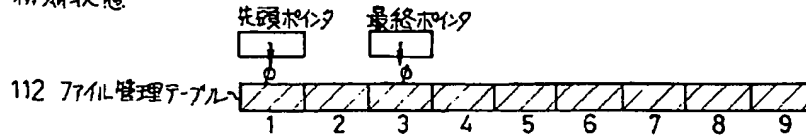
削除順序問い合わせ／設定フラグ
の内容の一例を示す図

| 意 味 | 値 |
|-------|---|
| 問 合 せ | 2 |
| 設 定 | 3 |

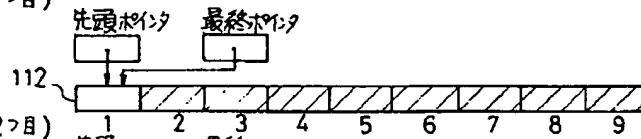
【図5】

図3,図4のフローにおける先頭及び最終ポイントの変化の様子を模式的に示した図

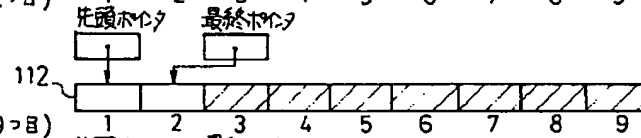
(a) 初期状態



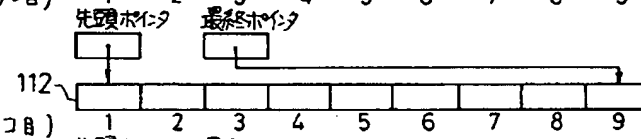
(b) ファイル作成時(1回目)



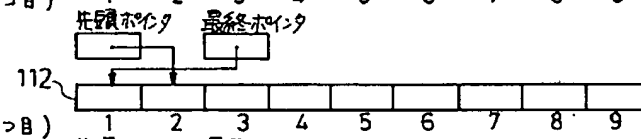
(c) ファイル作成時(2回目)



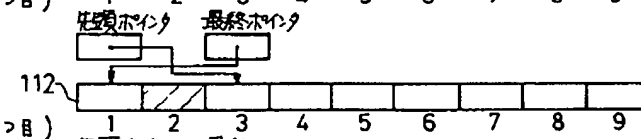
(d) ファイル作成時(9回目)



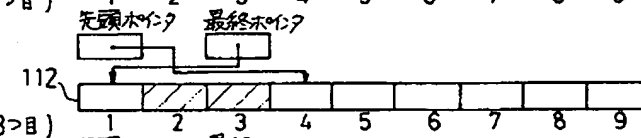
(e) ファイル作成時(10回目)



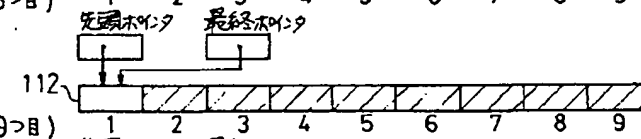
(f) ファイル削除時(1回目)



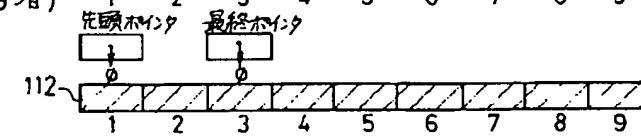
(g) ファイル削除時(2回目)



(h) ファイル削除時(8回目)

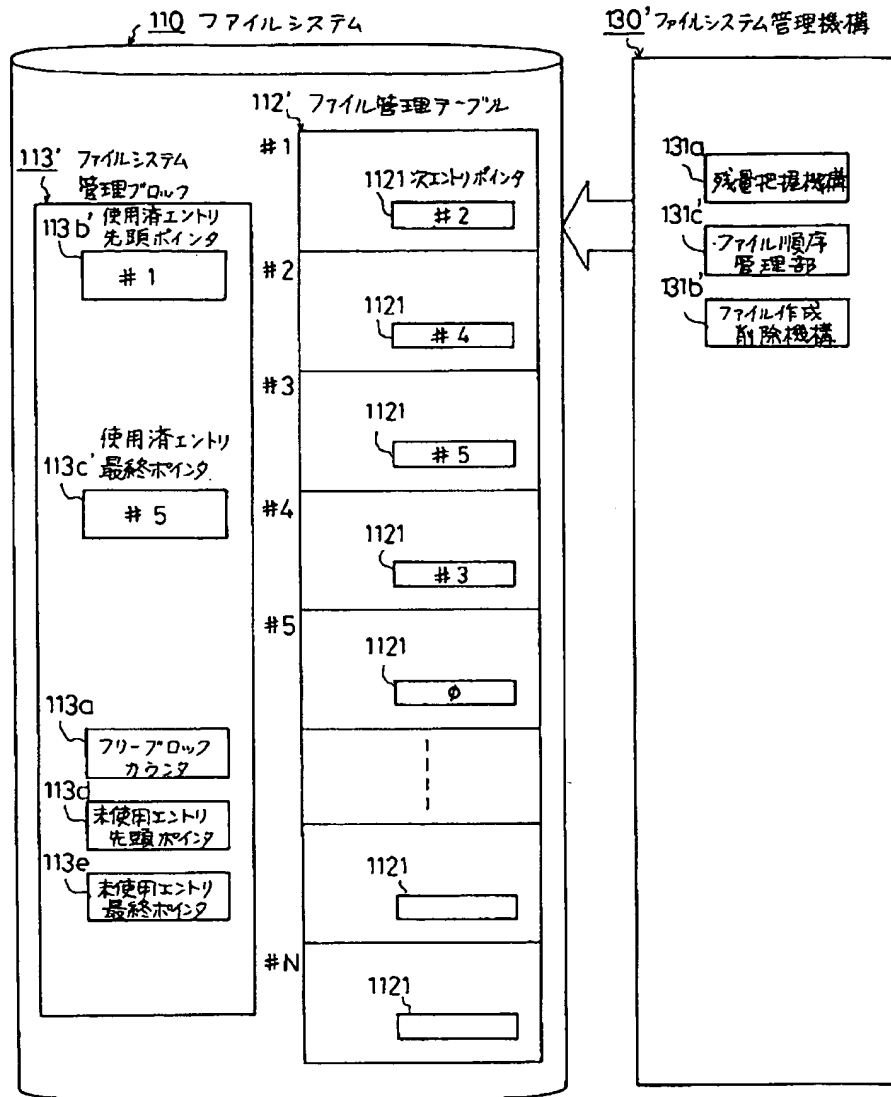


(i) ファイル削除時(9回目)

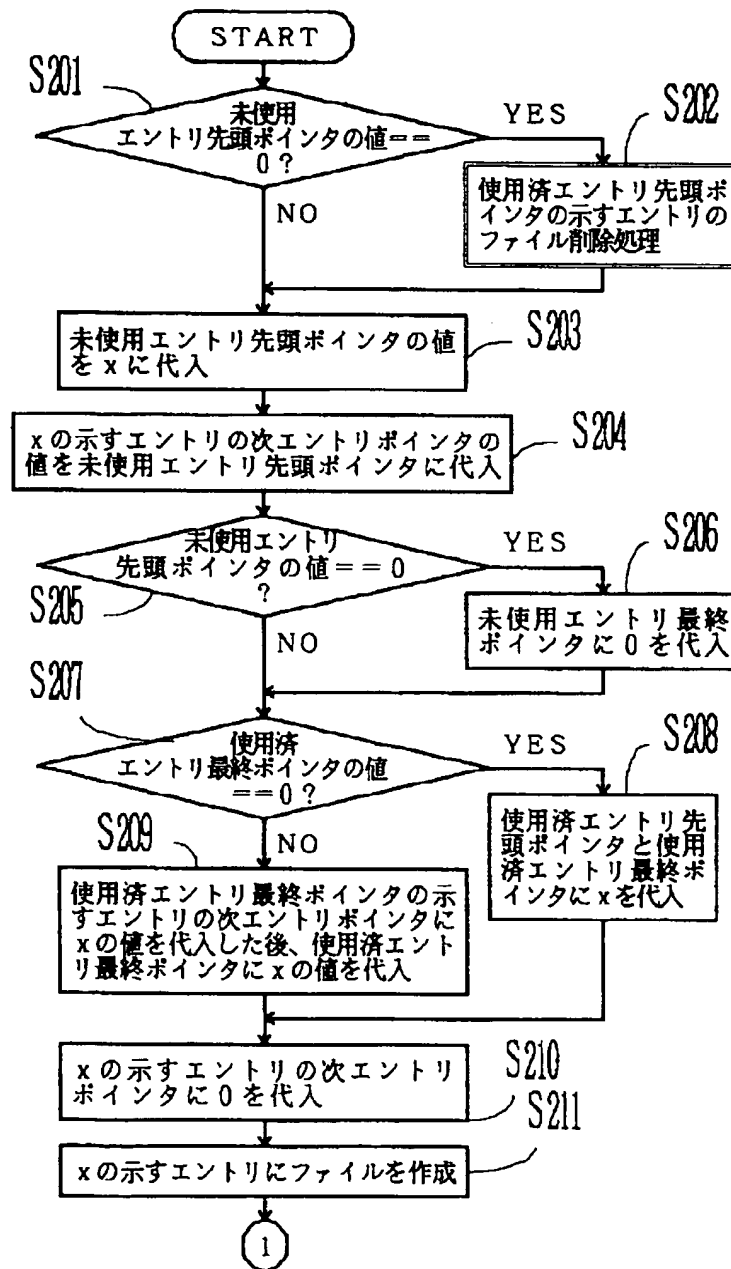


【図6】

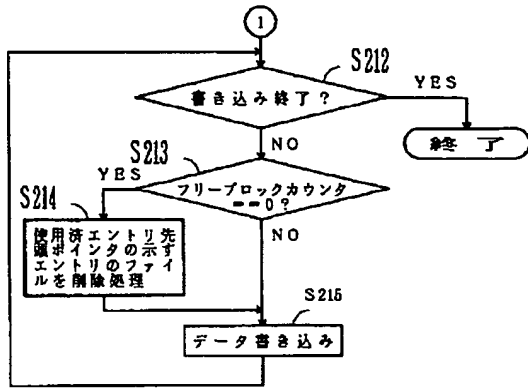
本発明の第一の基本構成の第二の実施例を示すブロック図



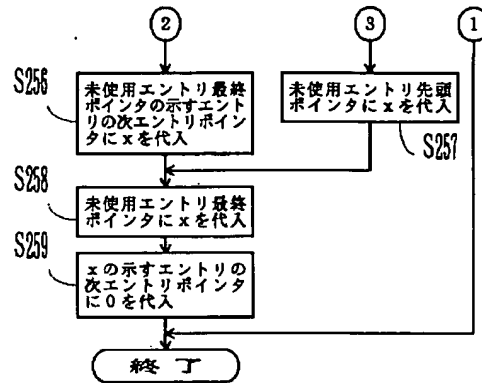
【図7】

図6のファイル順序管理部によるファイル
作成処理を説明するフローチャート(その1)

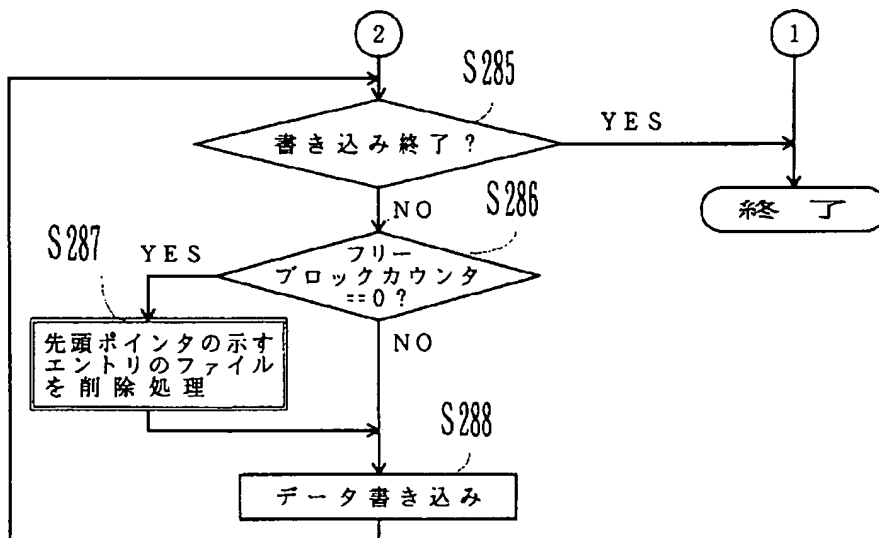
【図8】

図6のファイル順序管理部によるファイル
作成処理を説明するフローチャート(その2)

【図11】

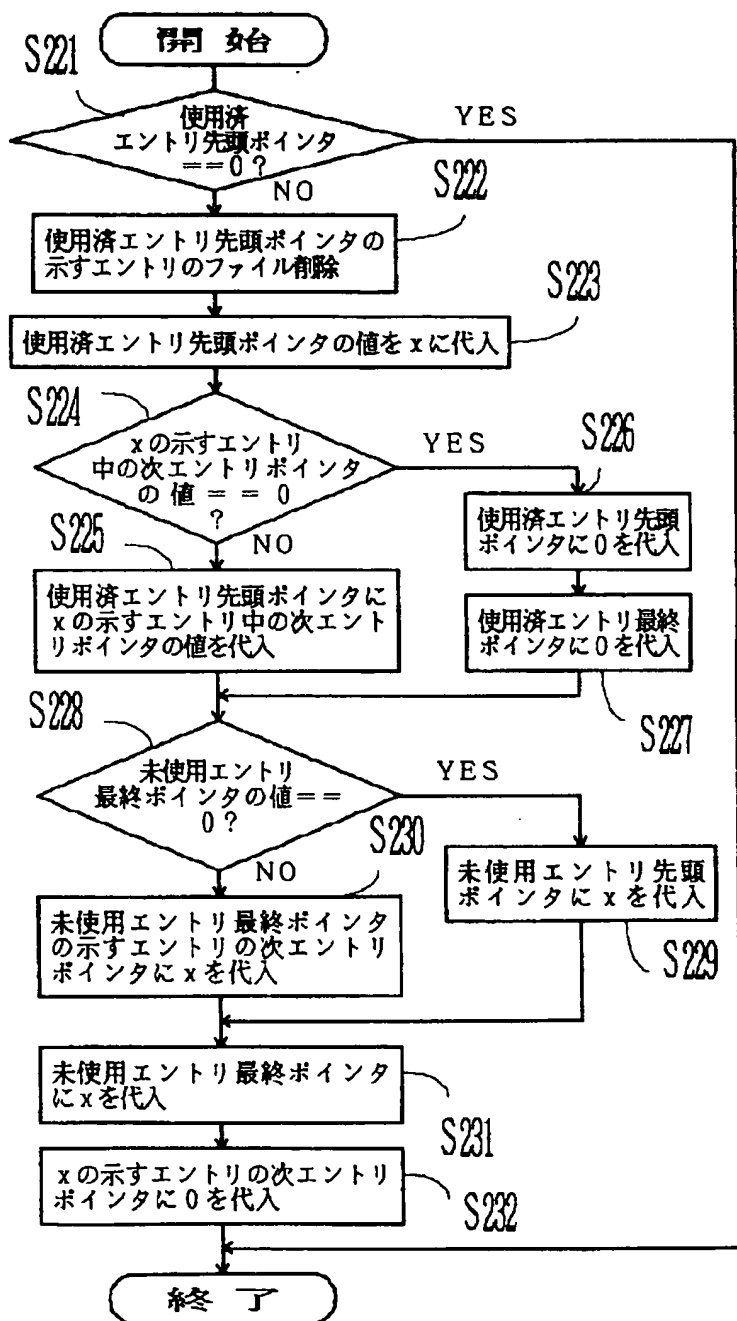
図6のファイル順序管理部による第2のファイル
削除処理を説明するフローチャート(その2)

【図13】

図6のファイル順序管理部によるファイル
更新処理を説明するフローチャート(その2)

【図9】

図6のファイル順序管理部による第1のファイル
削除処理を説明するフローチャート



【図28】

削除条件フラグ(mode)の一覧を示す図

| 意味 | 値 |
|--------|---|
| 作成日時 | 1 |
| 更新日時 | 2 |
| アクセス日時 | 3 |
| サイズ | 4 |
| ユーザ名 | 5 |
| グループ名 | 6 |
| リンク数 | 7 |
| タイプ | 8 |

【図30】

削除条件フラグ(cond)の一覧を示す図

| 意味 | 値 |
|----|---|
| EQ | 1 |
| GT | 2 |
| LT | 3 |
| GE | 4 |
| LE | 5 |
| NE | 6 |

【図35】

ファイルの削除順序情報の
優先順位問い合わせ/設定フラグ
を説明する図

| 意味 | 値 |
|-------------|---|
| 問合せ | 6 |
| 設定 | 7 |
| 削除順序情報数の問合せ | 8 |
| 削除順序情報数の設定 | 9 |

【図10】

図6のファイル順序管理部による第2のファイル
削除処理を説明するフローチャート(その1)

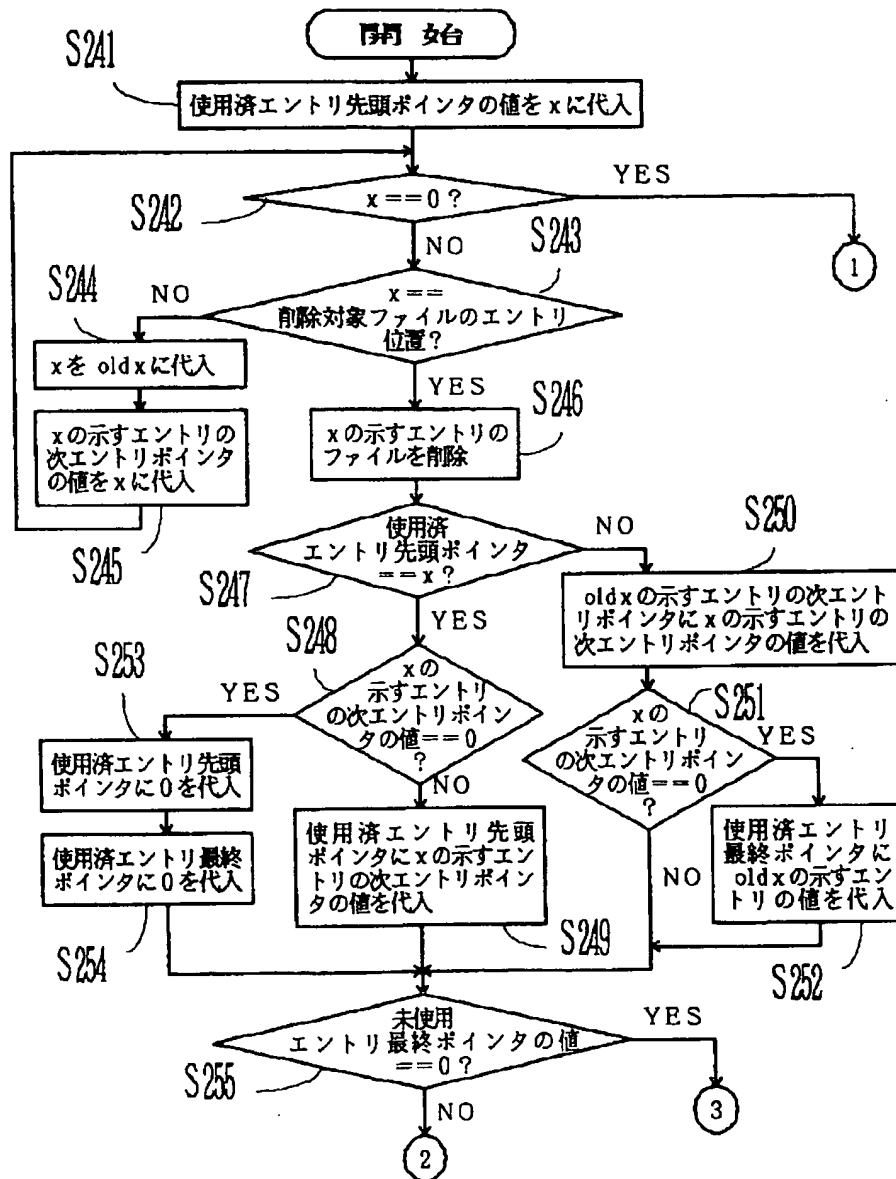
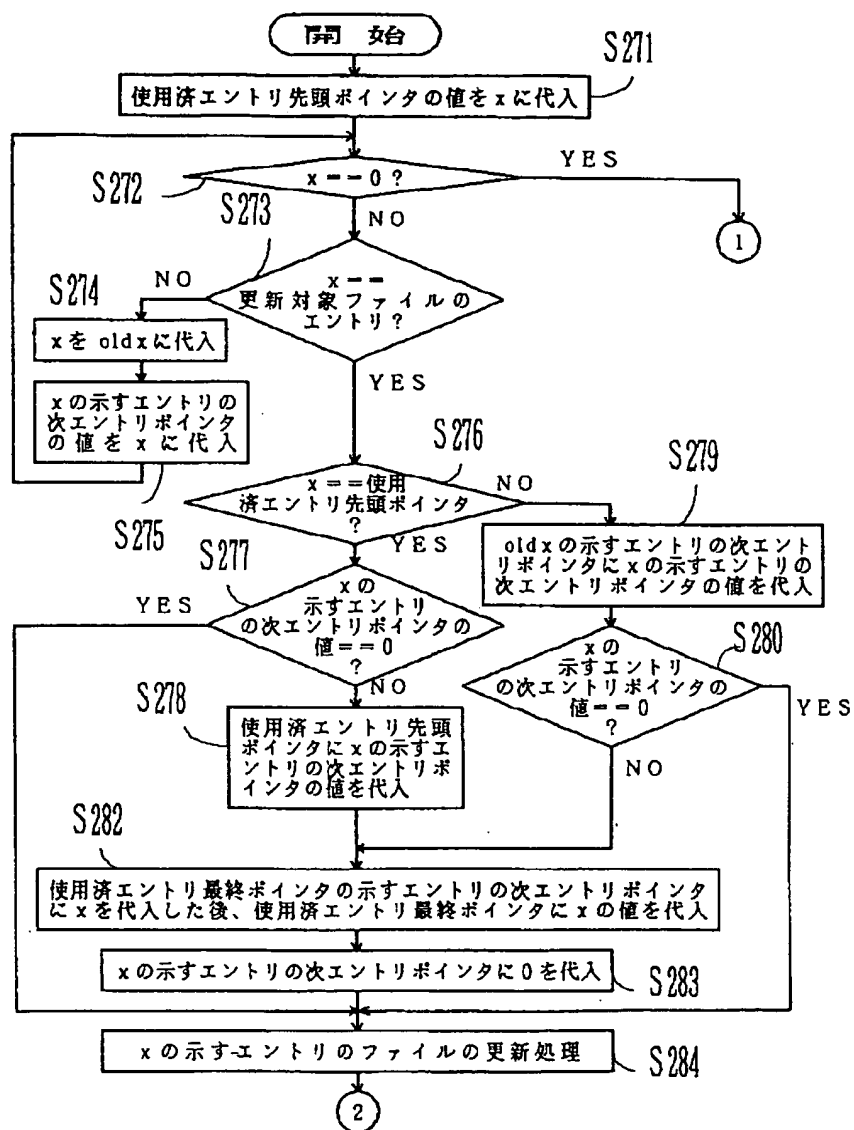


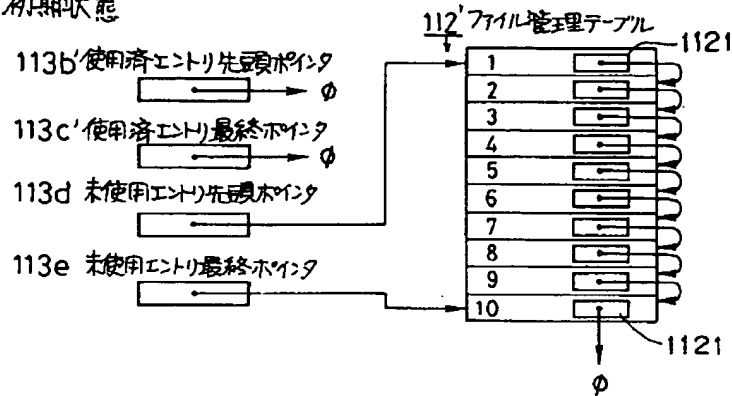
図6のファイル順序管理部によるファイル更新処理を説明するフローチャート(その1)



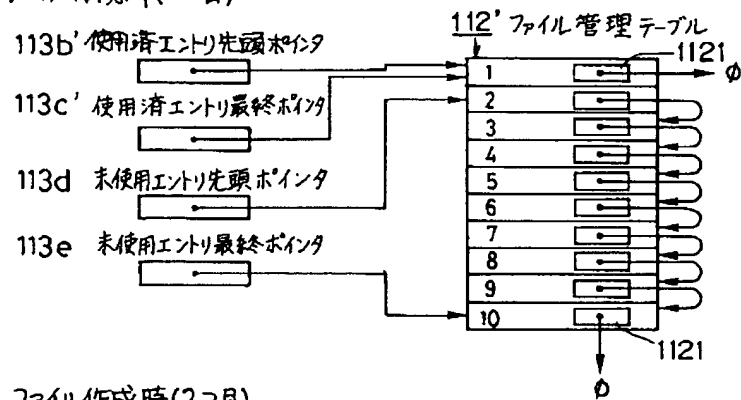
【図14】

図7～図13における各ポインタの状態を模式的に示した図

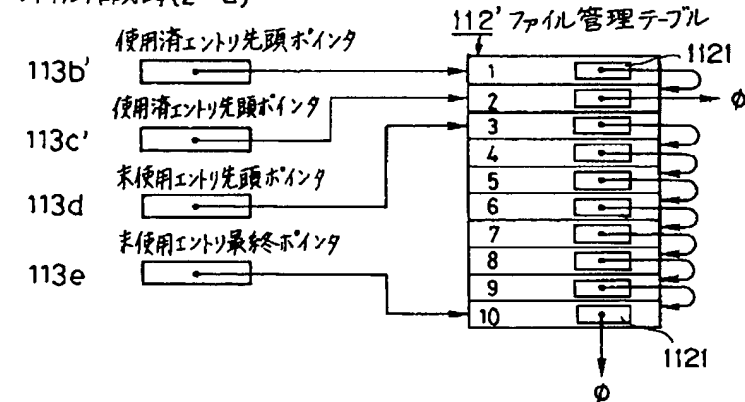
(a) 初期状態 (その1)



(b) ファイル作成時(1つ目)



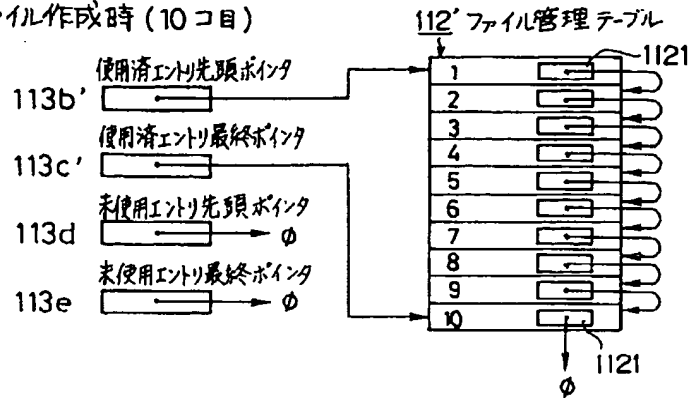
(c) ファイル作成時(2つ目)



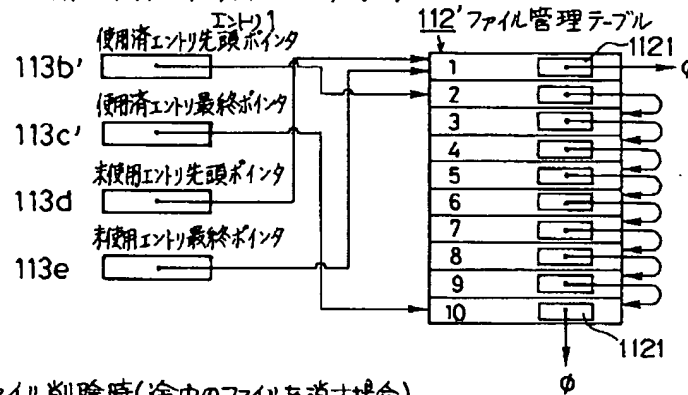
【図15】

図7～図13における各ポインタの状態を模式的に示した図

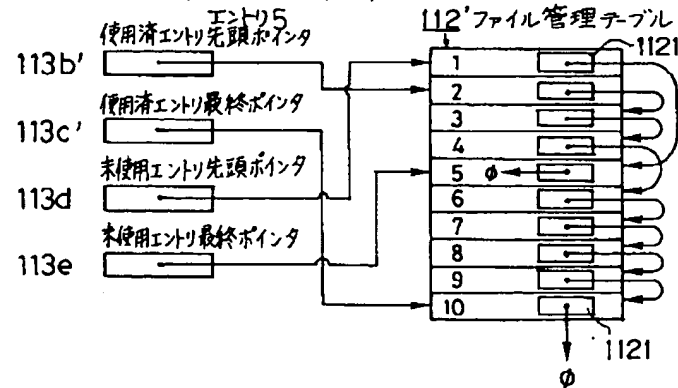
(d) ファイル作成時(10コ目) (その2)



(e) ファイル削除時(先頭のファイルを消す場合)

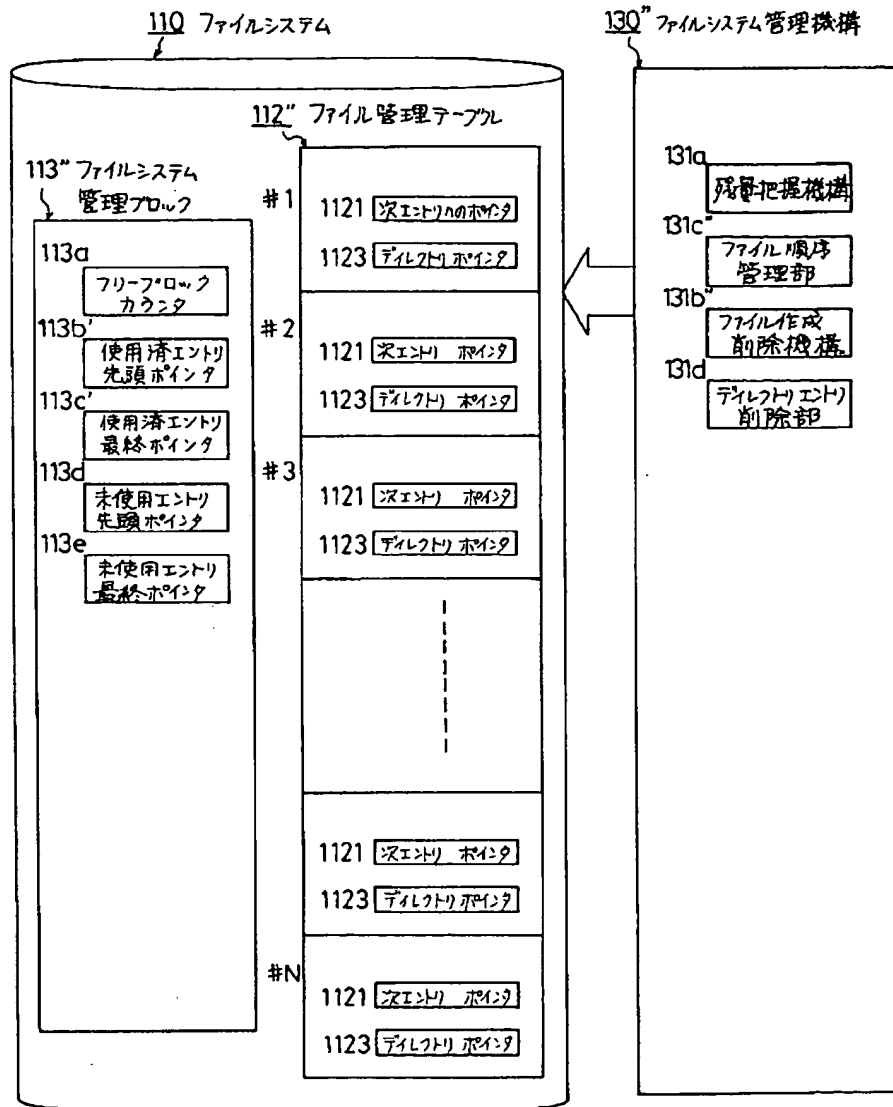


(f) ファイル削除時(途中のファイルを消す場合)



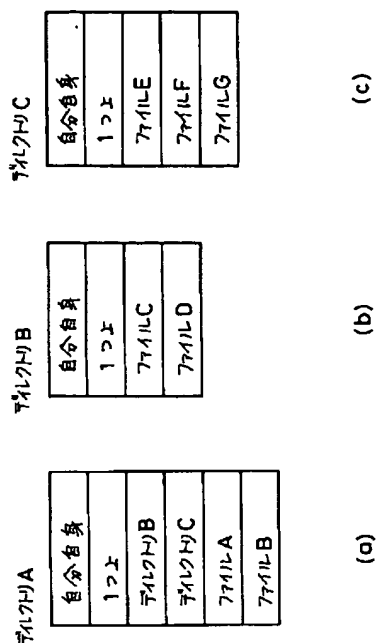
【図17】

本発明の第一の基本構成の第三の実施例を示すブロック図



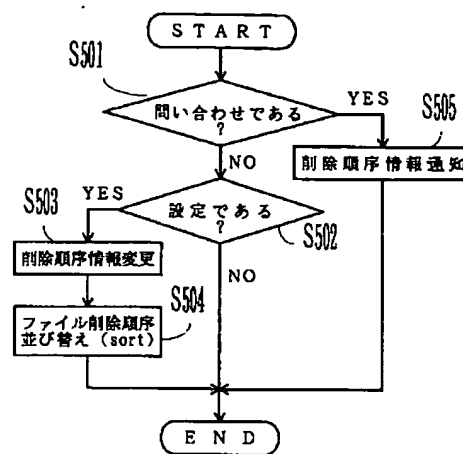
【図19】

ファイルシステムのデータブロック内に作成される
図18のディレクトリ構成に関する情報のデータ
構造を示す図



【図25】

削除順序問い合わせ/設定部の動作の
一例を説明するフローチャート



【図27】

削除順序情報管理部が
削除条件情報格納部に登録する
ファイルの削除順序情報の一覧を示す図

| 意味 | 値 (8進数) |
|---------|---------|
| 作成日時順 | 01 |
| 更新日時順 | 02 |
| アクセス日時順 | 04 |
| サイズ順 | 010 |
| アクセス回数順 | 020 |
| 逆順 | 0200 |

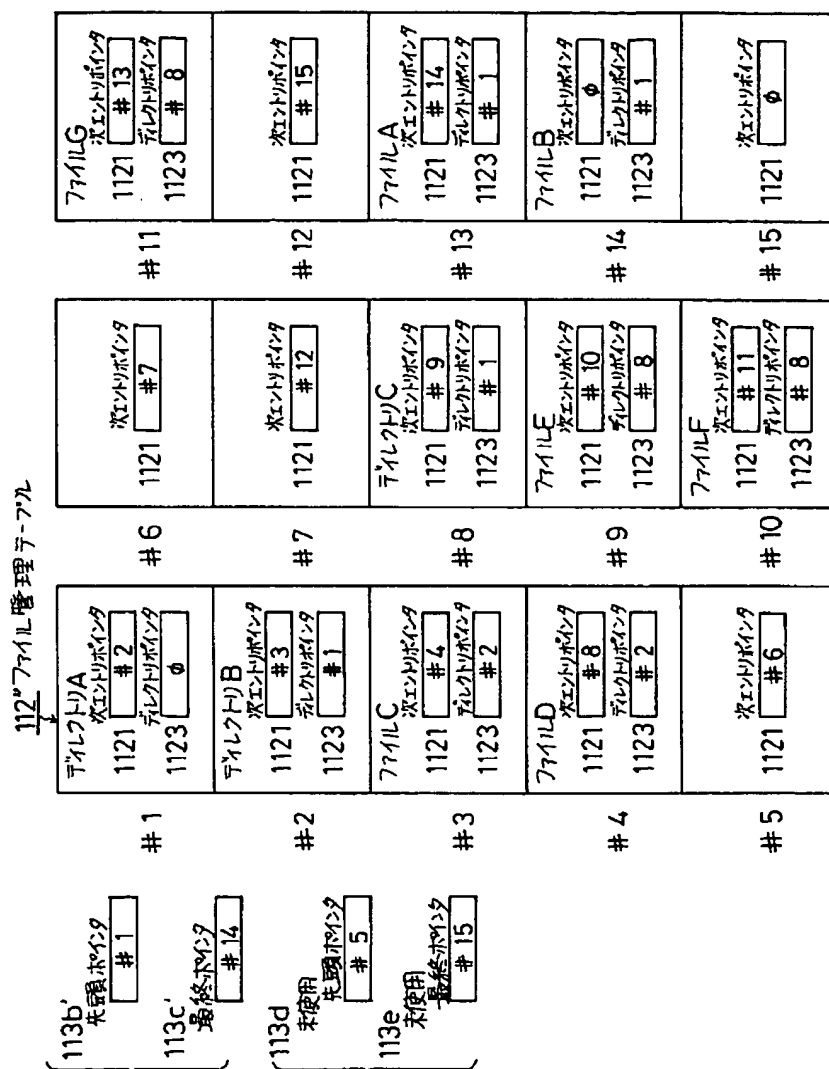
【図29】

削除条件フラグ(type)の一覧を示す図

| 意味 | 値 |
|--------------------|---|
| 通常ファイル | 1 |
| ディレクトリ | 2 |
| キャラクタ スペシャルファイル | 3 |
| ブロック スペシャルファイル | 4 |
| シンボリック リンクファイル | 5 |

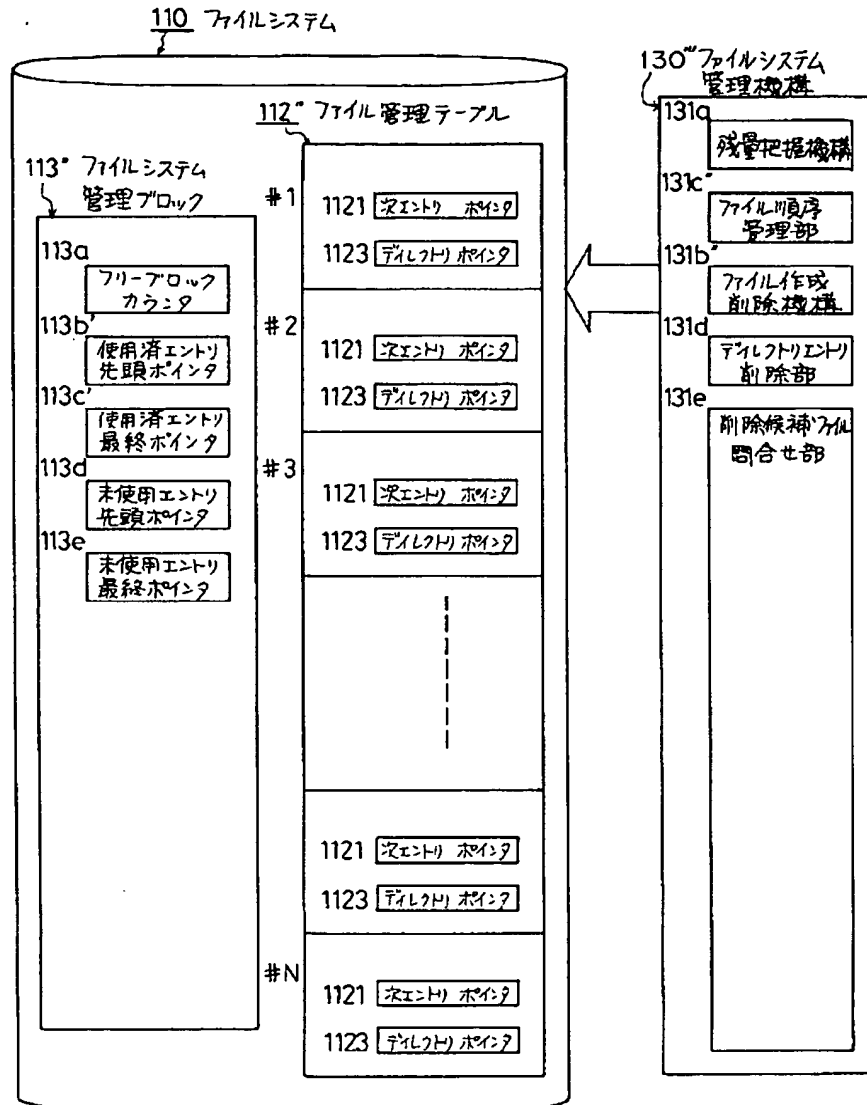
【図20】

図19に示すディレクトリのデータ構造を管理している
ファイルシステム内のファイル管理テーブルと
ファイルシステム管理ブロックの構成図



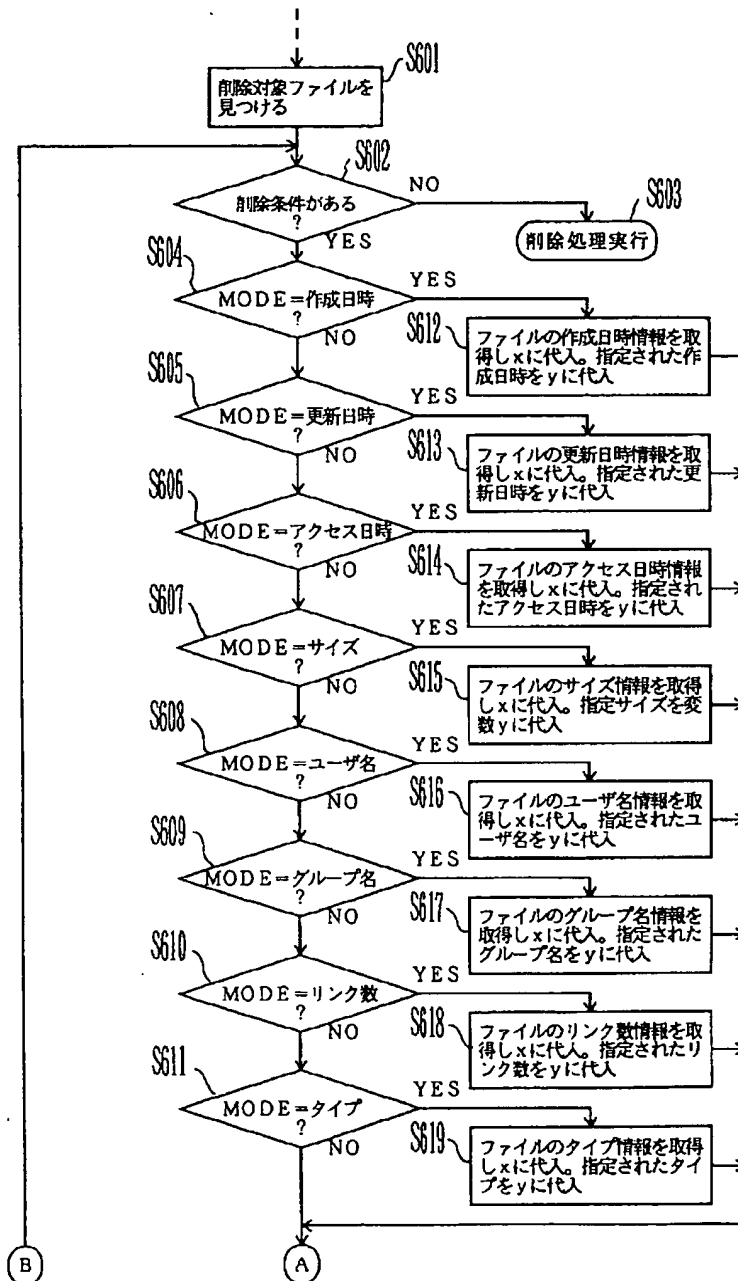
【図22】

本発明の第一の基本構成の第4の実施例のシステム構成図



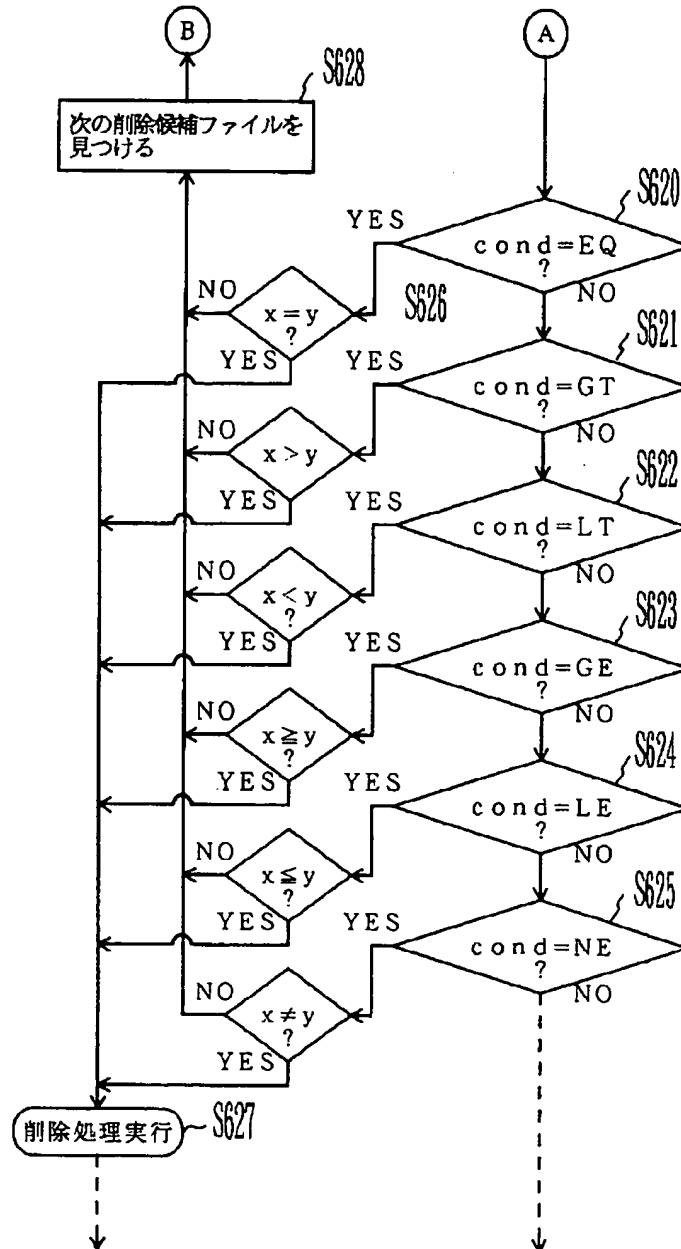
【図32】

ファイル削除部が削除条件情報格納部に登録されているファイルの削除条件情報を考慮して、ファイルを削除する処理の概略を説明するフローチャート(その1)



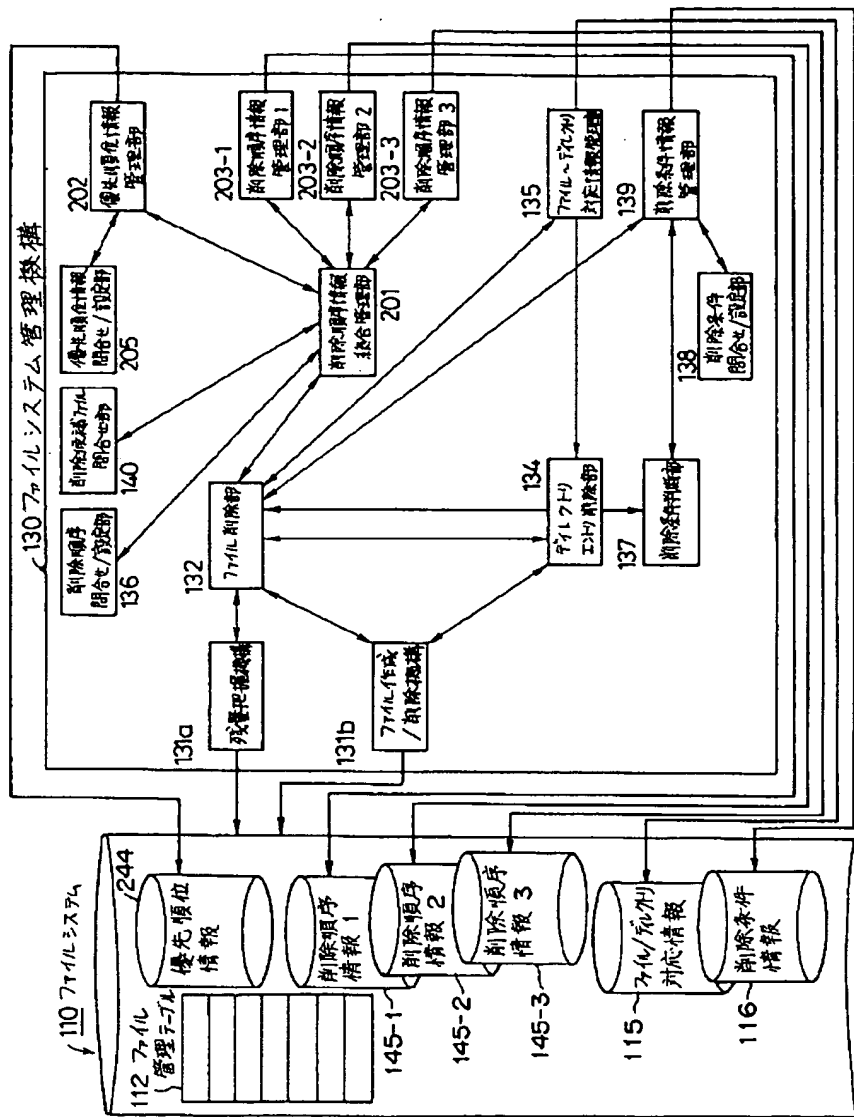
【図33】

ファイル削除部が削除条件情報格納部に登録されている
ファイルの削除条件情報を考慮して、ファイルを削除
する処理の概略を説明するフローチャート（その2）



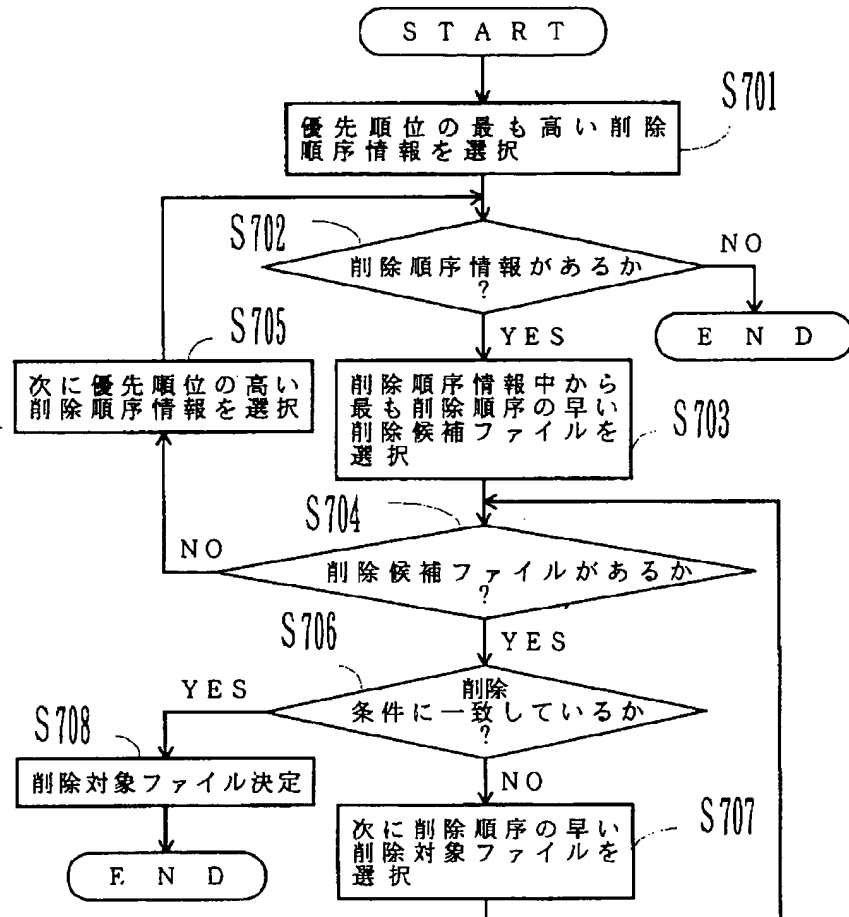
【図34】

本発明の第二の基本構成の
システム構成を示すブロック図



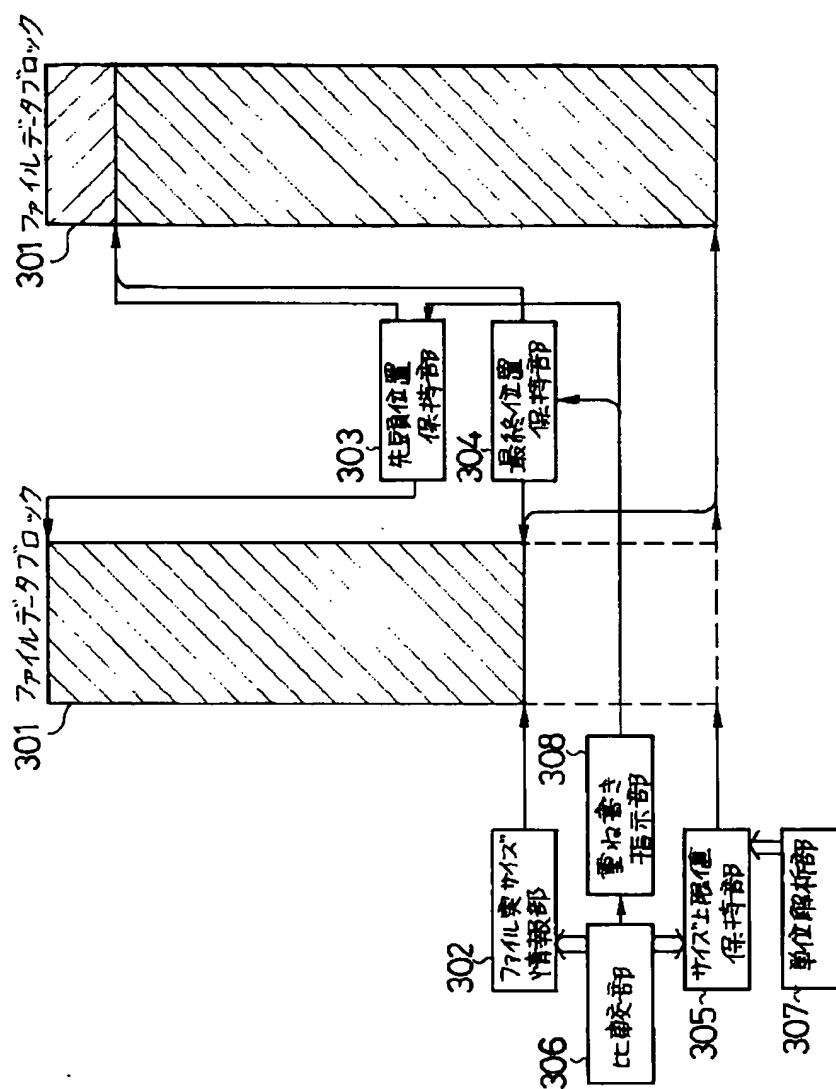
【図36】

優先順位を考慮した削除候補ファイル検索処理を示すフローチャート



【図37】

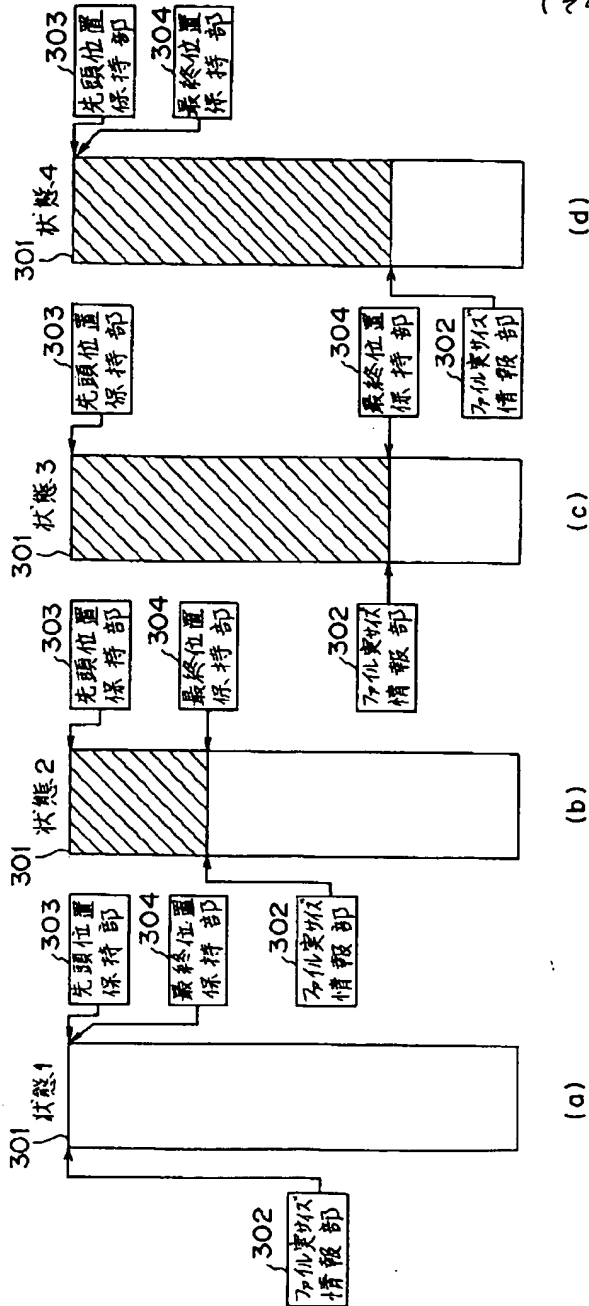
本発明の第七の実施例のシステム構成を
示すブロック図



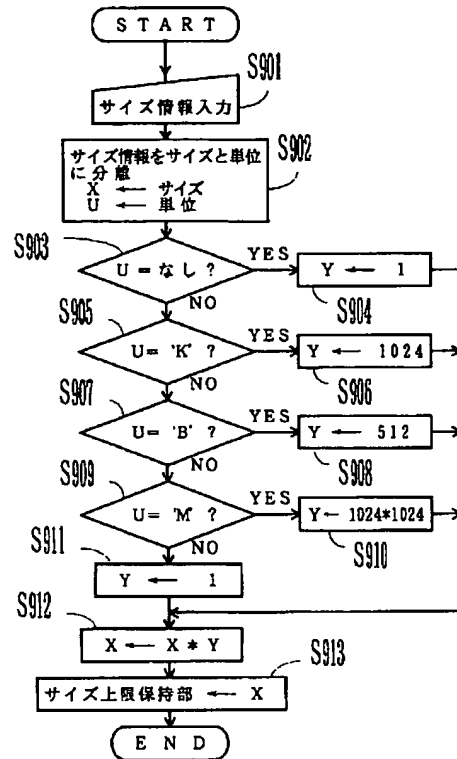
【図38】

第七の実施例の第一の動作を説明する図

(その1)



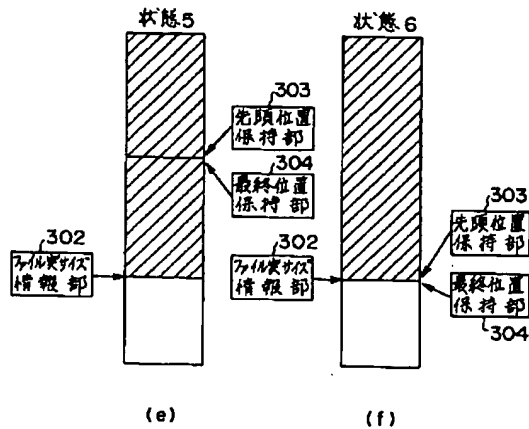
【図43】

単位解析部によって実行される第七の実施例
の第三の動作の処理を説明するフローチャート

【図39】

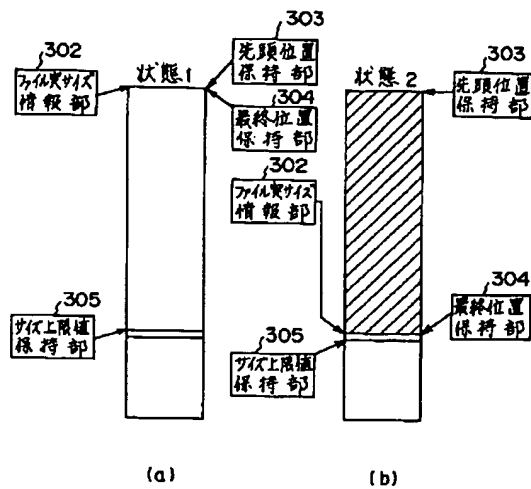
第七の実施例の第一の動作を説明する図

(その2)



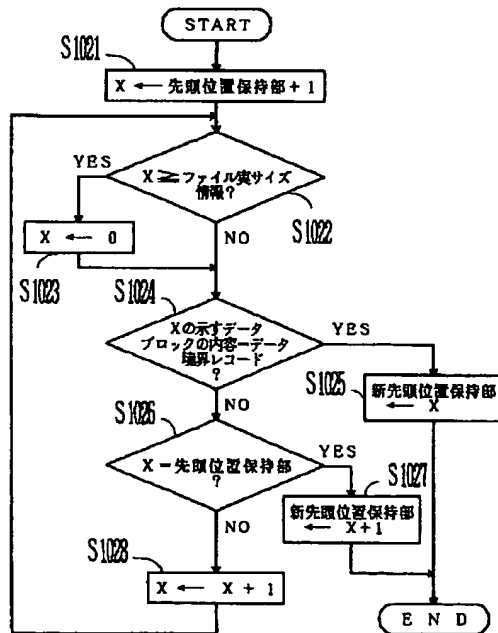
【図40】

第七の実施例の第二の動作を説明する図



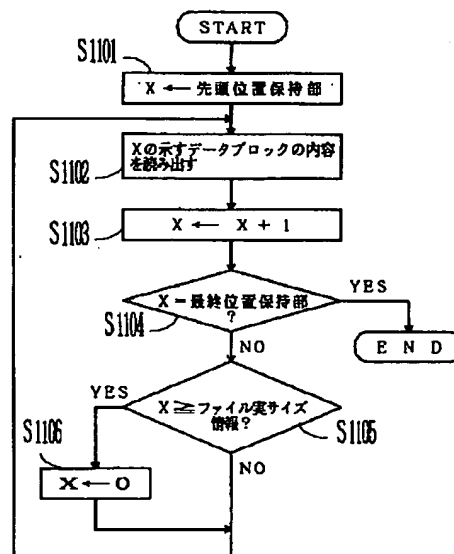
【図48】

データ境界検出部によって実行される図44の
ステップにおいて、先頭位置保持部に設定する
新たな先頭位置を見つける処理を説明する
フローチャート



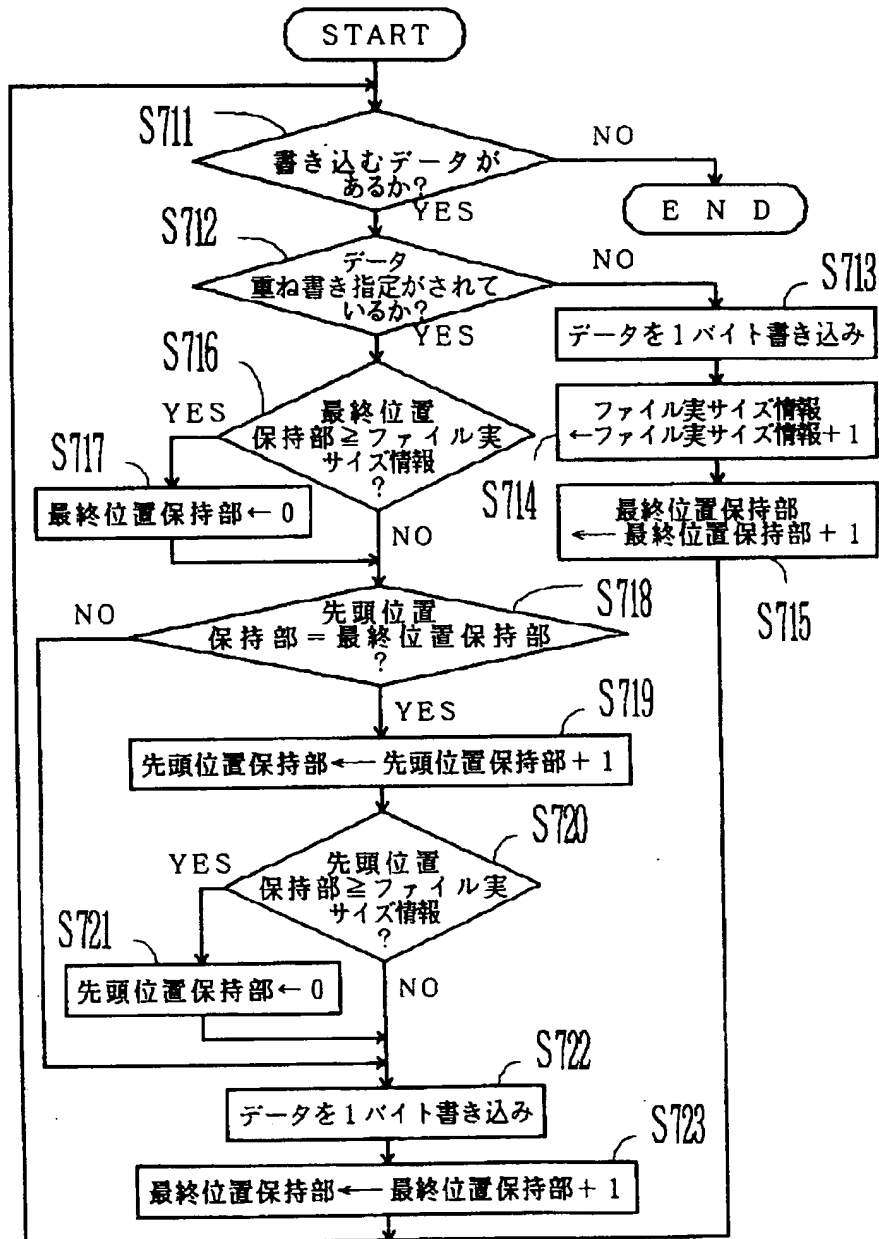
【図49】

ファイルデータブロックからデータを読み出す場合の
処理を説明するフローチャート



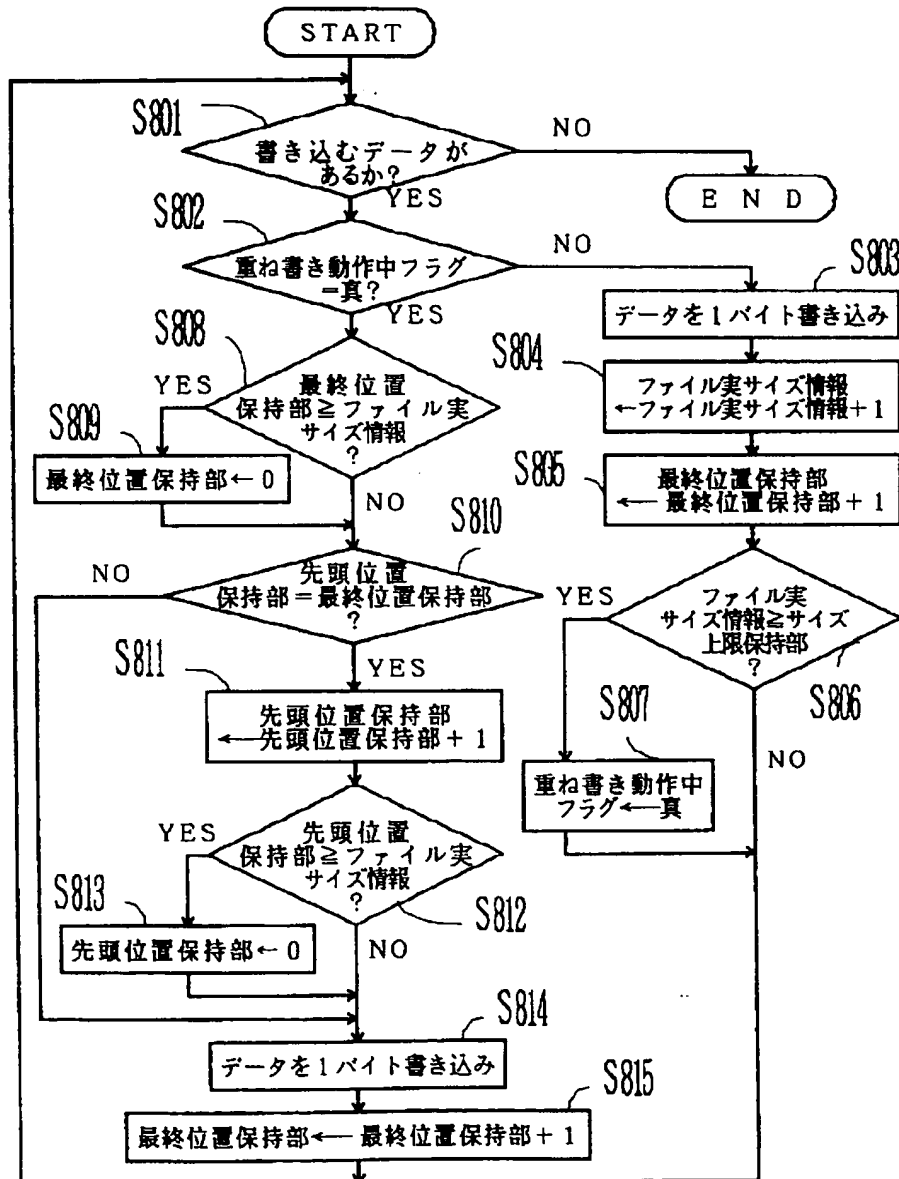
【図41】

データ書き込み部によって実行される
第七の実施例の第一の動作の
処理を説明するフローチャート



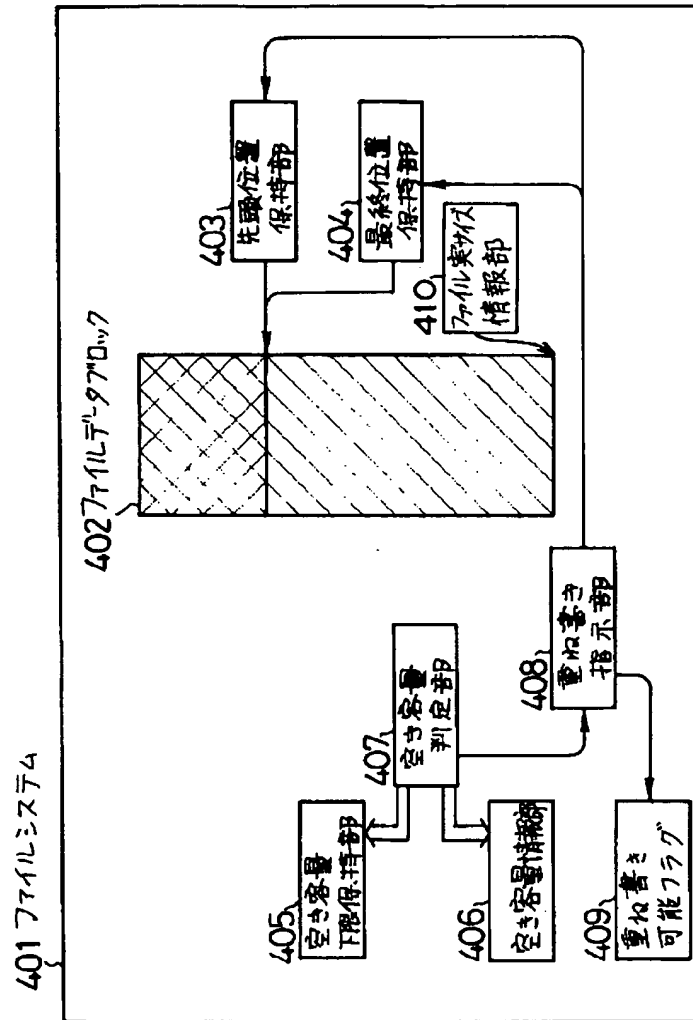
【図42】

データ書き込み部によって実行される第七の実施例の
第二の動作の処理を説明するフローチャート



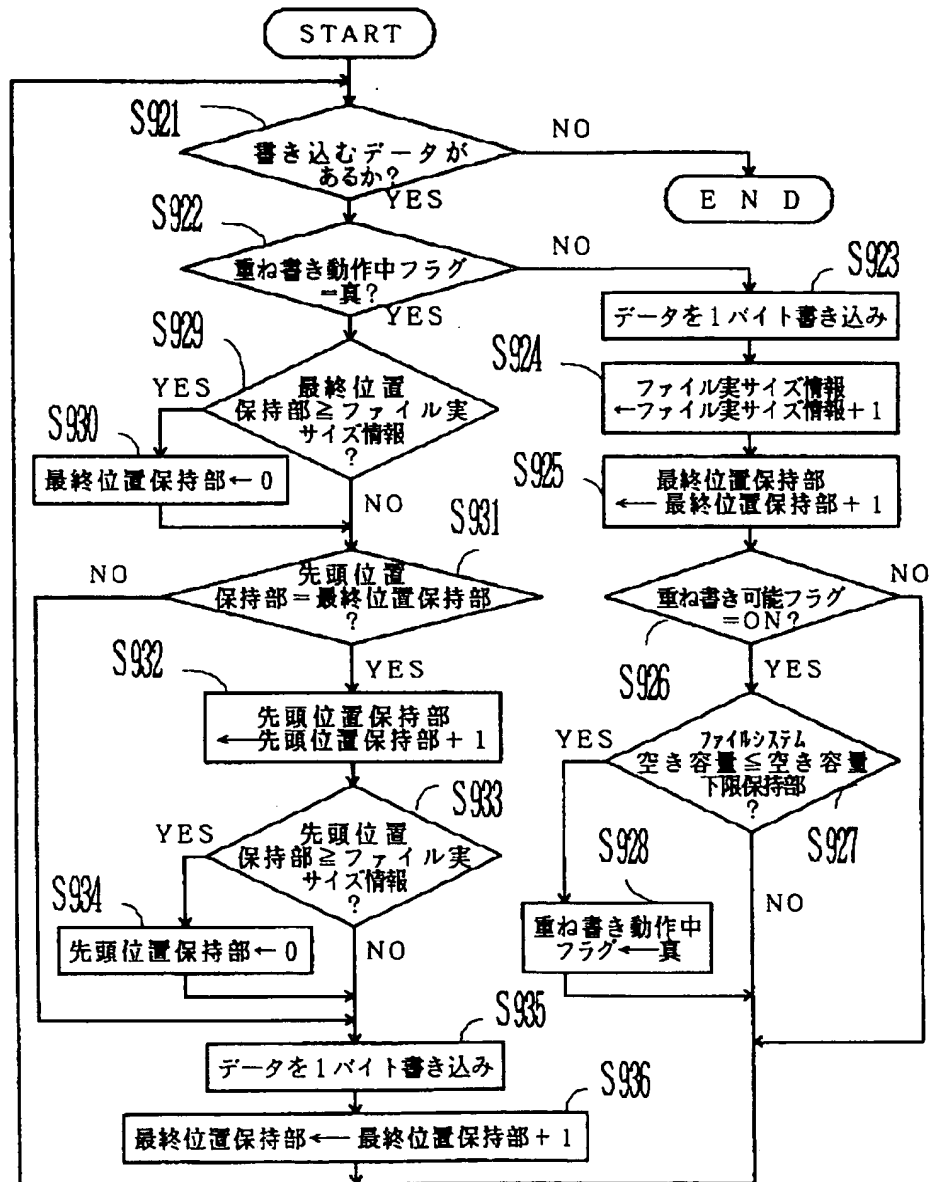
【图44】

本発明の第八の実施例のシステム構成図



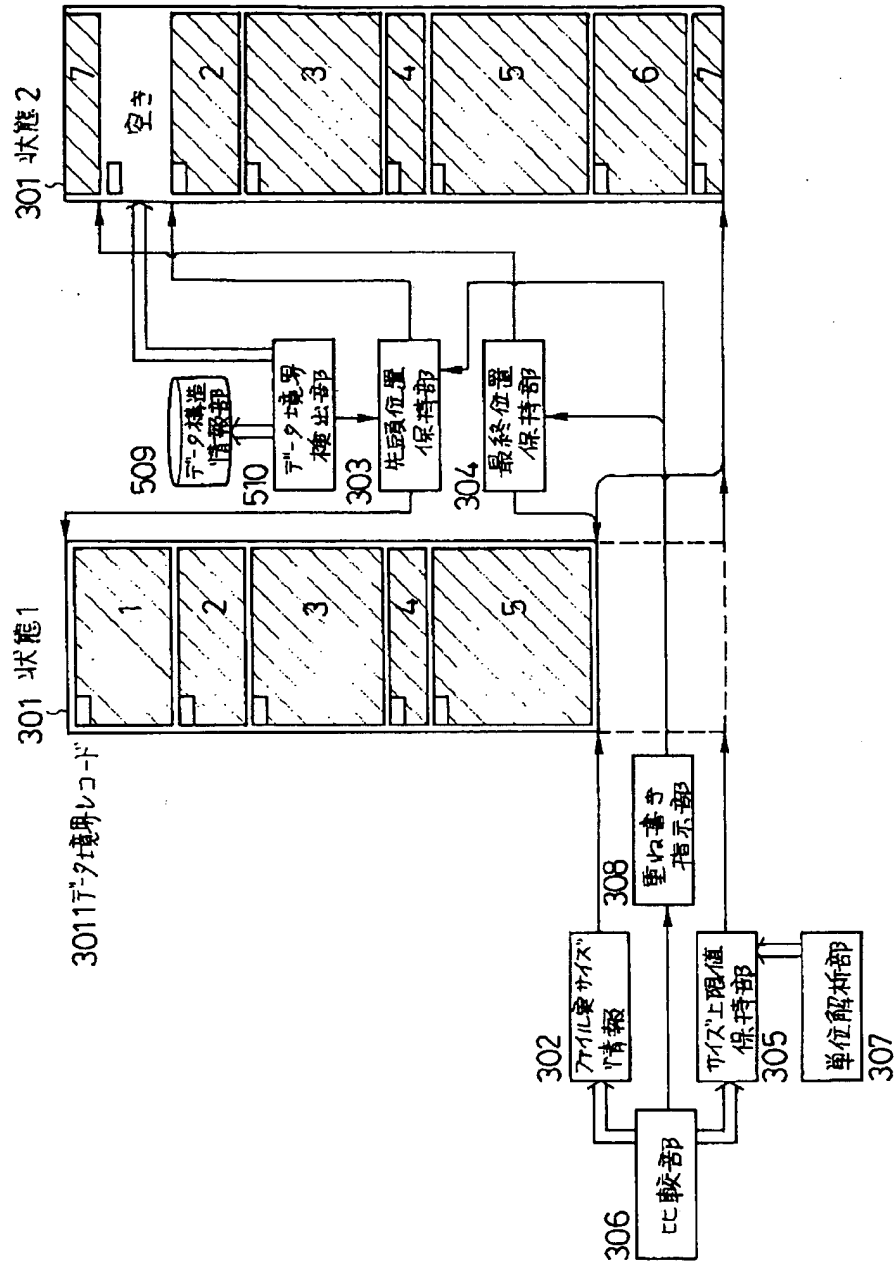
【図45】

データ書き込み部の動作を説明するフローチャート



【図46】

本発明の第九の実施例のシステム構成図



【図47】

本発明の第八の実施例における
データ書き込み部の動作を説明するフローチャート

